Journal of New Music Research - Special Issue on Music and Machine Learning - 40 (2) 2011

"Copy and Scale" Method for Doing Time-Localized M.I.R. Estimation: Application to Beat-tracking

Geoffroy Peeters
Ircam Sound Analysis/Synthesis Team - CNRS STMS
1, pl. Igor Stravinsky - 75004 Paris - France
peeters@ircam.fr

Abstract

In this work we propose a "copy and scale" method based on a Nearest Neighbor paradigm to estimate time-localized parameters and apply it to the problem of beat-tracking. The Nearest Neighbor algorithm consists in assigning the information of the closest item of a pre-annotated database to an unknown target. It can be viewed as a "copy and paste" method. The "copy and scale" method we propose consists in "scaling" this information to adapt it to the properties of the unknown target. In order to represent time-location, we represent the content of an audio signal using a sampled and tempo-normalized complex DFT of its onset-energy-function. This representation is used as the code over which the Nearest Neighbor search is performed. Along each code of the Nearest Neighbor space, we store the corresponding annotated beat-marker positions in a normalized form. A search is then performed for a set of tempo assumptions. Once the closest code and best tempo assumption are found, the normalized beat-markers of the closest item are scaled to this tempo in order to provide the estimation of the beat-markers of the unknown item. We perform a preliminary evaluation of this method and show that, with such a simple method, we can achieve results comparable to the ones obtained with sophisticated approaches.

1 Introduction

Music Information Retrieval from audio signal can be roughly divided into two categories: - estimation of global parameters (global meaning that the parameters is applicable to the whole file duration, an example of this is the music genre) - and estimation of local parameters (local meaning that the parameters is time-localized, examples of this are beat/downbeat, onset or pitch; which take place at specific time positions).

Problems of the first category are usually solved using machine-learning approaches including the K-Nearest Neighbor (K-NN) method. For K equal 1 (Nearest Neighbor method) the method can be viewed as a "copy and paste" method, where the parameters (for example the music genre) of an unknown item are estimated by "copying and pasting" the parameters of the closest item of a pre-annotated database. Problems of the second category are usually solved using signal processing algorithms without the use of machine-learning techniques. In this work we propose a "copy and scale" method based on a Nearest Neighbor paradigm to estimate local (time-localized) parameters and apply it to the problem of beat-tracking.

Suppose we have a very large database of audio-items, each one of them has been annotated into beat positions. Suppose we want to estimate the beat positions of an unknown audio extract. The usual process is to run a beat-tracking algorithm on it. However, one may think of using an audio fingerprint technique to look if this extract is present in the pre-annotated database, get the precise time position of it and then simply "copy and paste" the annotated beat-markers of the database to the unknown item. However, this would require a very large database and require that the unknown audio extract is part of the database items. However, instead of using an audio fingerprint technique (which implies an exact match of timbre, rhythm, harmony, instrument and production), we relax the code to only highlight one of the specific aspects of the content. For example, we define a code such that the distance between two audio items is small when they have the same rhythmic pattern, tempo and are time-aligned. Then any item in the database with a very small distance to the unknown item can be used to provide the beat-markers of it; even if this item is different from the unknown item. The required database may still be very large. We further relax the constraint on the code to provide small distances when the "rhythmic patterns" are close; this, independently of the tempi and time-alignments. Then the required database can be much smaller (since we only require it to represent the diversity of rhythmic patterns). Of course, because the matching is not anymore complete (maybe the closest item has a different tempo and/or alignment) it may be necessary to re-aligned and re-scale the beat-markers before copying them. The code must provide the necessary information for this.

1.1 Paper content and organization

Starting from this, we propose in this work, a method that allows applying a Nearest Neighbor (NN) approach for the estimation of time-localized information and apply it to the case of beat-marker estimation. Instead of the usual "copy and paste" approach underlying the NN approach, we propose a "copy and scale" approach.

The above-mentioned distance between two items is obtained by coding the items using a sampled and tempo-normalized complex DFT of their onset-energy-function. This code is said to be tempo-independent, since any two audio items with similar rhythmic pattern but different tempo will have the same code (see [29] for more details). This is obtained by normalizing the frequencies of the DFT by the one of the tempo. The inclusion of the DFT phase part provides the necessary information for time-alignment between any two sequences (through phase relationships). We describe this representation in part 2.1 and 2.2. A NN database of annotated audio item is created. For each annotated item, we store its coded representation, which will be used to perform the search. Along each code we store the time-localized annotations which will be "copied and scaled" to provide the estimation of the unknown item parameters. For this, we store the item's tempo, rhythm class[1] and beat-marker positions in a time-normalized form. We explain this in part 2.3. For a known item, the code corresponding to the annotated tempo is computed (part 2.2.1). For an unknown item, a set of codes corresponding to a set of tempo assumptions are computed (part 2.2.2). For each code of this set, we perform a search in the NN database using a complex distance. We describe this in part 2.4. The tempo assumption and the database-item providing the smallest distance are chosen to provide the estimation of the unknown item parameters: tempo, rhythm class and beat-markers. For this, the NN beat-markers are de-normalized according to the estimated tempo. This is the "scale" part of the "copy and scale". We describe this in part 2.5. In part 3, we propose optimizations of the method and provide figures related to its computation time. In part 4, we present the results of an evaluation on the "ballroom-dancer" test-set using a Leave-One-Out approach. The results obtained in this primary study show the applicability of the proposed approach. We finally conclude in part 5.

We summarize the various steps of the proposed method in Figure 1.

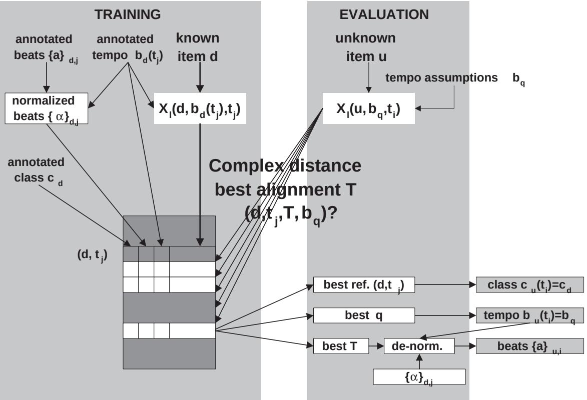

Figure 1: Training and evaluation part of the "copy and scale" method for tempo, class and beat-tracking estimation.

[1] In the following, the term "rhythm class" refers to the grouping of items into classes according to their temporal rhythmic pattern (such as the classes provided by the "ballroom dancer" test-set).

1.2 Motivations for the present work

There exist numerous beat-tracking and tempo-estimation methods, all based on sophisticated signal processing algorithms. Failures to detect correctly the tempo and beat markers are mainly caused by 1) weak onsets, 2) time-variable tempo and 3) complex rhythmic patterns (such as in Latin or African music). The third cause is due to the lack of knowledge of these patterns. It is of course possible to introduce this knowledge on a case by case. The proposed approach based on a Nearest Neighbor (NN) paradigm allows to introduce this knowledge on a large scale without explicit models (NN does not use any models).

1.3 Related works

We summarize here the main trends in the fields of tempo-estimation, beat-tracking and rhythm classification. We refer the reader to [12] or [25] for more details on recent approaches and/or results.

Tempo estimation algorithms can be first classified according to the analyzed materials: - symbolic data or - audio data. Algorithms based on audio data analysis usually start by a front-end which either - plays the role of an “audio-to-symbolic” translator [22] [11], - or extracts frame-based audio features such as energy or energy variations [32] [26] [23]. Depending on the kind of information provided by this front-end and the context of the application, a large variety of processes are used to track/estimate the tempo: - time interval histograms [4] [15], - periodicity measure (Fourier transform, auto-correlation function, narrowed-ACF, wavelets, comb filter-bank). The periodicity measure is then used - to estimate directly the tempo - or to serve as observation for the estimation of the whole metrical structure through (probabilistic) models [23] [11] [24]. Some authors also propose the use of templates for tempo estimation - in the time/phase domain [24] [23] [34], - or in the spectral domain [27].

Beat-tracking methods can be roughly classified according to the front-end of the model: - discrete onset representation [11], - or continuous-valued onset function [32] [23] [3]. They can also be classified according to the model used for the tracking: - multi-agents model [11] [5], - use of resonating comb-filers [32] [21], - probabilistic formulations [23] [1] [18] [24] [8] [3]. Recent approaches succeed to use directly the phase information to derive beat-phases [7] [17].

For **rhythm classification**, the proposed methods mainly differ on: - the type of information being represented (event positions, acoustical characteristics of the events, or both), - and the way they are represented (sequence of events, histogram, profiles, evolution). Foote [10] proposes the use of a beat spectrum. Tzanetakis [33] proposes the use of a beat histogram from which various features are derived. Paulus [26] models the rhythm characteristics as a sequence of audio features and uses DTW to compute the distance between two sequences. Gouyon’s [13] tests a set of 73 features derived from the tempo, a periodicity histogram and the Inter-Onset-Interval Histogram to characterize the rhythm. Dixon [6] adds to Gouyon features a representation of the temporal rhythmic patterns derived from the energy evolution of the signal inside each bar. Holzapfel [19] proposes the use of Dynamic Periodicity Warping (DPW) to compute rhythmic similarity; or in [20], the use of the Melin Transform (MT) to provide a scale and tempo independent rhythm representation.

Works which are the most related to our work, are the followings. In [9], Eronen proposes to use a database of templates and a K-NN-regression to find the best tempo of an unknown signal. While we also use templates, those are complex-valued in our case. While we also use K-NN, our tempo assignment method is very different. Finally, [9] does not deal at all with time-localized information such as beat-markers. In [29], we show that rhythm classification can be achieved with a high accuracy using solely the observation of the normalized amplitude DFT of an onset-energy-function. We will use here the complex DFT. In [17], Grosche proposes to use the phase of DFT spectrum to derive the beat-positions. [17] doesn’t use templates or machine learning[2]. We didn’t find any previous work concerning the use of Nearest Neighbor for beat-tracking, or concerning its use in the complex domain. We therefore think that our proposed approach is novel.

[2] Although not directly related to our approach, we also mention the work of Gouyon [14] who use machine-learning to classify signal-frames into beat and non-beat classes.

2 Proposed approach

2.1 Complex spectral representation

In our method, each audio item is represented by a complex spectral representation which is used as the search code. The flowchart of its computation is indicated in Figure 2.

For a given audio item, we first extract an onset-energy-function $o(n)$, representing at each time n the likelihood of an onset, using the method explained in [27]. This function has a sampling rate of 200.45Hz. We perform a Short Time Fourier Transform analysis of $o(n)$ using a rectangular window of 8s duration[3] with a hop size of 1s. We denote it by $X_k(o, t_i)$, where i is the frame index and k the index of the Fourier frequencies f_k. Considering a tempo frequency $b(t_i)$ (expressed in Hz) over time t_i, we sample the complex spectrum $X_k(o, t_i)$ at the frequencies $f_k = b(t_i) \cdot f_l$ with $f_l = \{\frac{l}{4} : 1 \leq l \leq 32\} \cup \{\frac{l}{3} : 1 \leq l \leq 24\}$. These frequencies represent the harmonic series corresponding to a 4/4 meter $\left(\frac{b(t_i)}{4}\right)$ and a 3/4 meter $\left(\frac{b(t_i)}{3}\right)$ up to $8b(t_i)$. We denote by $X_l(o, b(t_i), t_i)$ the 48-dimensional complex vector representing time t_i for a tempo $b(t_i)$. $X_l(o, b(t_i), t_i)$ is made amplitude independent by normalizing it by its maximum value over l. The onset-energy-function $o(n)$ around time t_i is therefore represented by a sum of L complex components:

$$\hat{o}(n) = \Re\left(\sum_l X_l(o, b(t_i), t_i) e^{j\Omega_l \frac{n}{sr}}\right) = \sum_l A_l(o, b(t_i), t_i) \cos\left(\Omega_l \frac{n}{sr} + \Phi_l(o, b(t_i), t_i)\right) \quad (1)$$

where we denote by A_l and Φ_l the modulus and phase of the complex X_l, by $\Omega_l = 2\pi b(t_i) f_l$ the frequencies in radian and by sr the sampling rate (200.45Hz in our case). The three main advantages of using this complex representation are: 1) the representation is compact: an 8s signal (1600 samples) is represented using only 48*2 values, hence the storage in the NN database is reduced; 2) the use of the phase part allows to represent the time-location of the events occurring in $o(n)$, hence it allows us to perform alignment of two codes; 3) it allows a better modeling of the signal $o(n)$ than the one obtained using an amplitude-only based model (using only A_l without phase)[4].

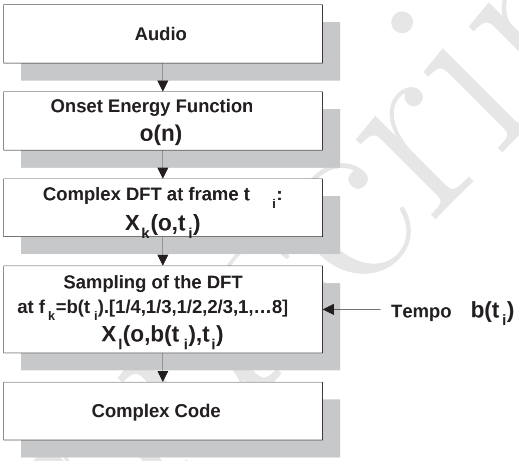

Figure 2: Flowchart of the computation of the complex spectral representation used as the search code.

It should be noted that the code X_l does not contain information about tempo. This code only depends on the rhythmic pattern of the audio item. Therefore if Track-A and Track-B have the same rhythmic patterns (but different tempo or starting time) then Code-A and Code-B will be equal. It should be noted also that the position of the beat-markers are specific to the rhythmic pattern, their inter-distances depend on the tempo and the start of their sequence depends on the signal relative position. Therefore, if Code-A=Code-B (i.e. the two tracks have similar rhythmic patterns), then we can use the beat-markers of A to get the ones of B. We need of course to re-align them to synchronize the respective start of their sequences and to re-scale them to their corresponding tempi.

2.2 Item representation

The representation of a given audio item using the proposed complex-code necessitates the knowledge of the tempo $b(t_i)$. We here distinguish the computation of the code • for known-items (the ones used

[3] The window length is chosen in order to achieve good spectral resolution between the harmonics of the bar frequency for tempi down to 60bpm in a 4/4 meter.

[4] If we define the modeling error as $\epsilon = [\sum_n (o(n) - \hat{o}(n))^2]/[\sum_n o(n)^2]$, the average ϵ obtained on the test-set of part 4 is • $\epsilon = 0.4187$ using X_l and • $\epsilon = 1.6223$ using A_l.

for the creation of the NN database) for which the tempo is known (from annotation); • for unknown-items (the items for which we want to estimate the parameters) for which the tempo is unknown (it is one of the parameters to estimate).

2.2.1 Known item representation

For a given known audio item d, annotated into tempo over time $b_d(t_j)$, we compute the complex code using $b(t_j) = b_d(t_j)$ to yield $X_l(d, b_d(t_j), t_j)$. For each frame j of each item d, we store the 48-dimensions complex vector $X_l(d, b_d(t_j), t_i)$ in the NN database.

2.2.2 Unknown item representation

In order to compute the complex code of an unknown audio item (therefore with unknown tempo), we make a set of tempo assumptions $b_q \in \{B\}$, where $\{B\}$ is the set of tempo assumptions. For each of these tempo assumptions we compute the complex code using $b(t_i) = b_q$ to yield $X_l(u, b_q, t_i)$. If there are Q different tempo assumptions, we compute Q different representations $X_l(u, b_q, t_j)$. The set of tempo assumptions can be taken by sampling the frequencies between a minimum and a maximum tempo frequency. For each of these tempo assumptions b_q, we will compare the corresponding code $X_l(u, b_q, t_i)$ to all the codes contained in the NN database. The b_q leading to the closest NN item will define the best tempo for the unknown item.

2.3 NN database construction

For a given audio item d at frame t_j, annotated into tempo over time $b_d(t_j)$, rhythm class c_d and beat positions $\{a\}_d$, we store in the NN database - the 48-dimensions complex-code $X_l(d, b_d(t_j), t_j)$ - the corresponding annotated tempo $b_d(t_j)$, - the rhythm class c_d and - the sub-set of *normalized beat positions* $\{\alpha\}_{d,j}$.

2.3.1 Normalized beat positions $\{\alpha\}_{d,j}$

If we note s_j and e_j the starting and ending time of frame t_j, the subset $\{a\}_{d,j}$ of beat positions is made of the a_d for which $s_j \leq a_d$ and $a_d \leq e_j$. The normalized subset is then defined as $(\{a\}_{d,j} - s_j) \cdot b_d(t_j)$ and is noted $\{\alpha\}_{d,j}$. It represents the beat markers of item d at frame j for a normalized beat frequency of 1Hz. This process is illustrated in the upper part of Figure 3.

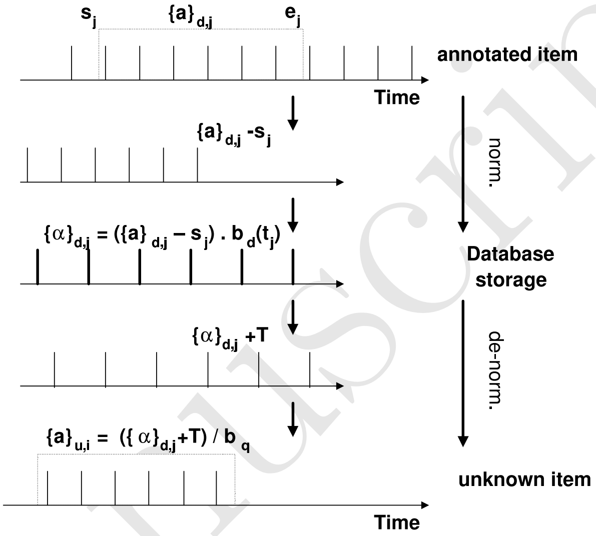

Figure 3: Computation of normalized (upper part) and de-normalized (lower-part) beat-markers

Algorithm 1 summarizes the various steps for the NN database construction. The variable definitions are summarized in Table 1.

Algorithm 1 NN database construction

```
for all known items $d$ of known rhythm class $c_d$, known tempo $b_d(t_j)$ over time-frame $t_j$ and known beat-markers $\{a\}_d$ do
    for all frame $t_j$ do
        Compute the complex spectrum $X_k(d, t_j)$ at frame $t_j$
        Compute the complex code $X_l(d, b_d(t_j), t_j)$
        Compute the normalized beat-marker positions $\{\alpha\}_{d,j} = (\{a\}_{d,j} - s_j) \cdot b_d(t_j)$
        Add a new entry in the NN database with $X_l(d, b_d(t_j), t_j)$, $c_d$, $b_d(t_j)$ and $\{\alpha\}_{d,j}$
    end for
end for
```

2.4 Search over the NN database

In order to estimate the parameters of a frame t_i of an unknown item u we perform a Nearest Neighbor search. Since (u, t_i) is represented by Q complex code $X_l(u, b_q, t_i)$ (representing the Q tempo

Variable	Definition
$o(n)$	**o**nset-energy-function over sample n
u	an **u**nknown item
d	a known item of the NN **d**atabase
$X_k(o, t_i)$	complex spectrum of $o(n)$ around time t_i and at frequency k
f_k	frequency [in Hz]
t_i,t_j	time of frame i, j [in seconds]
s_i,e_i	starting and ending time of frame i
$X_l(o, b, t_i)$	sampled and tempo-normalized complex template considering a tempo b and a frame t_i
f_l	normalized frequencies: $f_l = \{\frac{l}{4} : 1 \leq l \leq 32\} \cup \{\frac{l}{3} : 1 \leq l \leq 24\}$. The frequencies f_k of X_k which are sampled to create X_l are: $b \cdot f_l$
$l \in [1, 48]$	index of normalized frequencies
$b(t_i)$	tempo at time t_i [in Hz]
$b_d(t_i)$	annotated tempo at time t_i
$b_q \in \{B\}$	one tempo assumption [in Hz]
$\{B\}$	the set of tempo assumptions
c_u, c_d	class of item u, of item d
$\{a\}_d$	beat-marker positions of item d [in seconds]
$\{a\}_{d,i}$	sub-set of $\{a\}_d$ for markers belonging to the frame t_i
$\{\alpha\}_{d,i}$	normalized version of $\{a\}_{d,i}$

Table 1: Variable definitions for the "copy and scale" algorithm.

assumptions b_q), we perform Q searches. Given that X_l is a complex code the search is performed using a distance in a complex space. For each search, we compute the distances between one of the Q complex code $X_l(u, b_q, t_i)$ and all the codes $X_l(d, b_d(t_j), t_j)$ of the NN database (representing all the frames j of all the items d). We then perform a Nearest Neighbor search[5].

In we denote by $U(l) = X_l(u, b_q, t_i)$ the code of the unknown item and by $D(l) = X_l(d, b_d(t_j), t_j)$ one of the database items, the Complex distance between U and D can be defined as

$$d_{CP}(U, D, T) = \sqrt{\sum_l d_l^2(U, D, T)} \qquad (2)$$

where

$$\begin{aligned} d_l^2(U, D, T) &= A_U^2(l) + A_D^2(l) \\ &- 2A_U(l)A_D(l)\cos(\Phi_U(l) - \Phi_D(l, T)) \end{aligned} \qquad (3)$$

where A_X represents the modulus of the complex X, Φ_X its phase, l the index in the complex vector, and T the best lag between the temporal signal $u(n)$ and $d(n)$ (corresponding to $U(l)$ and $D(l)$). The best lag is the one that minimizes the complex distance (maximizes the temporal synchronization). For this, each member T of a set of lags is tested and the phase spectrum of $D(l)$ modified according to $\Phi_D(l, T) = \Phi_D(l) - 2\pi f_l T$. Because U and D are independent of tempo, it is possible to compare two codes representing signal with different initial tempo (this wouldn't be possible using the correlation between temporal sequences). Other advantages of this spectral computation (over a temporal correlation computation) are: - the possibility to pre-compute the phase increments (because the frequencies f_l are known in advance) - the possibility to give different weights to the various frequencies l in order to emphasize some of them.

Because the computation time of the complex distance is high, in part 4 we will test a configuration of our method in which a rough search is first made using an Euclidean or a One-minus-cosine distance, then the fine search using the Complex distance is only performed on the closet item. In this case the Euclidean d_E and the One-minus-cosine d_C distance between the modulus of U and D are defined by

$$\begin{aligned} d_E(U, D) &= \sqrt{\sum_l (A_U(l) - A_D(l))^2} \\ d_C(U, D) &= 1 - \frac{\sum_l A_U(l)A_D(l)}{\sqrt{\sum_l A_U^2(l)}\sqrt{\sum_l A_D^2(l)}} \end{aligned} \qquad (4)$$

2.5 Copy and scale the parameters

The result of the search over the NN database provides the reference to an item d, a frame j, a tempo assumption q and a lag T which minimize the distance to (u, t_i): $(u, t_i) \rightarrow (d, t_j, q, T)$.

[5] The method could be extended to a K-Nearest Neighbor search with $K > 1$. Because we did not find so far a proper way to deal with several results of beat-marker positions, we limit here our method to a Nearest Neighbor search.

Rhythm class estimation: We assign the rhythm class of the closet item d to the unknown item at frame t_i: $c_{u,t_i} = c_d$.

Tempo estimation: The tempo assigned to the unknown item at frame t_i is the best tempo assumption b_q (for the q minimizing the distance): $b_u(t_i) = b_q$[6].

Beat-position estimation: The normalized beat-markers $\{\alpha\}_{d,j}$ of (d, t_j) are used to get the beat-markers of (u, t_i). This is obtained after a de-normalization part, which is the "scale" part of our method. For the q and T minimizing the distance, the beat positions assigned to the unknown item at frame t_i are given by $\{a\}_{u,i} = (\{\alpha\}_{d,j} + T)/b_q$. This is illustrated in the lower part of Figure 3.

Algorithm 2 summarizes the various steps of the search over the NN database and of the "copy and scale" of the parameters. The variable definitions are summarized in Table 1.

Comments on time-varying tempo: In the case of time-varying tempo, computing the DFT of $o(n)$ is equivalent to compute the DFT of a signal with time-varying frequency. As explained in [31], the consequence of this is a widening of the main-lobes of the DFT amplitude and a curvature of the corresponding DFT phase. The proposed complex-code does not allow representing this and will provide the code corresponding to the average tempo over the analysis window. Also, the use of a single frequency $b_d(t_j)$ to sample the DFT and the beat-markers over the whole frame, does not take into account time-varying tempo. One possible solution to this, would be to reduce the length of the analysis window over which the DFT is computed in oder to approximate local parameter stationarity.

Algorithm 2 Search over the NN database and "copy and scale" of the parameters

```
for all unknown item $u$ do
  for all frame $t_i$ do
    Compute the complex spectrum $X_k(u, t_i)$ at frame $t_i$
    for all tempo assumptions $b_q \in \{B\}$ do
      Compute the complex code $X_l(u, b_q, t_i)$
      Search over the Nearest Neighbor database the closest $X_l(d, b_d(t_j), t_j)$
      Store the information of the closest item to $(u, t_i, b_q)$ and the best alignment $T$: $(u, t_i, b_q) \to (d, t_j, T)$
    end for
    Choose the minimum distance over the $b_q$.
    The results is: item $d$ at frame $t_j$ with lag $T$ with tempo assumption $b_q$
    assign class: $c_u = c_d$
    assign tempo: $b_u(t_i) = b_q$
    assign beat-markers: $\{a\}_{u,i} = (\{\alpha\}_{d,j} + T)/b_q$
  end for
end for
```

3 Implementation

3.1 Optimizations

A set of optimizations have been performed in order to reduce the computation time of the search.

1. For a given tempo assumption b_q, we only consider the items (d, t_j) of the NN database which have an initial tempo $b_d(t_j)$ close to b_q. The closeness is defined as $|\log_2(b_d(t_j)/b_q)| < 0.3785$[7]. The goal of this is not only to reduce the number of comparisons but also to avoid using codes of the NN database largely outside their initial context (defined by the tempo).

2. Since the computation time of the Complex distance is high, the NN search can be speeded up by performing it in two steps: 1) a rough search using an Euclidean distance or a One-minus-cosine distance (therefore considering only the amplitude part of the code); 2) a fine search over the closest item using the Complex distance to find the best alignment T between (u, b_q, t_i) and the top-ranked item (d, t_j). In part 4, we will compare the

[6] It should be noted that this a major difference with the method proposed by [9] which would have assigned a tempo value based on a regression over the tempi of the closest K-Nearest Neighbor items. During our experiments, we found that the choice $b_u(t_i) = b_q$ (our choice) leads to a 15% increase in tempo precision over the choice $b_u(t_i) = b_d(t_j)$ (which would have been Eronen's choice for the case $K = 1$).

[7] For example for $b_q = 100$, we consider all the items (d, t_j) which have a tempo ranging from 77 to 130 bpm

results obtained when using the Euclidean or One-minus-cosine distance before the Complex distance or when using solely the Complex distance.

3. In part 4, we will also compare the results obtained with two different set of tempo assumptions. The **first** set $\{B\}$ is the whole set of tempi. We test 321 candidate tempi representing all the possible tempi between 60 and 220 bpm with a step of 0.5 bpm. In this case, each frame t_i of an unknown item u is represented by $Q = 321$ complex vectors. The **second** set $\{B\}$ is a reduced set based on the output of a front-end tempo-estimation algorithm. The tempo estimation algorithm we used is the one described in [27]. We denote it by $\hat{b}(t_i)$. We then define $\{B\}_i$ as the set of typical octave errors of tempo-estimation algorithms (1/3, 1/2, 1, 2, 3 times the correct tempo). The reduced set at time t_i is therefore defined as $\{B\}_i = [1/3, 1/2, 1, 2, 3]\ \hat{b}(t_i)$. In this case, each frame t_i of an unknown item u is represented by $Q = 5$ complex vectors.

3.2 Computation time

For the evaluation of part 4, the number of items of the Nearest Neighbor database is 10470. We indicate here the computation time obtained using an Intel®Xeon®CPU 2.4GHz (only one processor used) with 24.6GB of RAM. The computation time depends on the configuration of the system (number of candidate tempi, distance used) and also depends on the distribution of the Nearest Neighbor database[8]. In Table 2, we indicate the average number of searches and the corresponding computation time for processing one frame (including search, alignment and scaling) for each configuration (choice of Q) and type of distance (using the Euclidean or One-minus-cosine distance before the Complex distance or using solely the Complex distance).

Q	Nb searches	Computation Time in seconds		
		Euclidean dist.	Cosine dist.	Complex dist.
1	4827	0.012 s	0.015 s	26.92 s
5	8480	0.021 s	0.026 s	39.89s
321	1 303 699	2.48 s	2.96 s	-

Table 2: Number of searches and computation time per frame for various configurations of the ”copy and scale” method.

4 Evaluation

In this part we evaluate the performances of the proposed approach for tempo estimation, rhythm classification and beat-tracking using various configurations of our system. It should be noted that the estimation of the three parameters are obtained at the same time based on the closest item found in the NN database.

4.1 Test-set

The evaluation is performed on the “ballroom dancer” test-set (as was used for the ISMIR2004 tempo induction contest) [16][9]. This test-set is often used for evaluation since it contains music for which the music genre and the rhythm class are closely related. It is composed of 698 tracks, each of 30 s long, representing the following music genre: ChaCha (111 instances), Jive (60), QuickStep (82), Rumba (98), Samba (86), Tango (86), Viennese Waltz (65) and Slow Waltz (110). Annotations into beat positions have been made by the author and have been cross-checked several times.

4.2 Evaluation rules

In order to evaluate the **classification performances**, we have used the global class accuracy (this is meaningful since the test-set is not highly unbalanced).

In order to evaluate the **tempo precisions**, we have used the measure proposed by [16], i.e. we measure the number of frames/ items for which the estimated tempo is within a 4% Tolerance Window of the annotated tempo. It should be noted that we do not consider octave detection as correct in this study[10].

[8] When using the first optimization, the number of searches depends of the tempo distribution of the NN database.

[9] The other MIREX test-sets are not available for testing outside the MIREX framework.

[10] Estimating one third, half, twice or three times the annotated tempo is not considered as correct in this study.

In order to evaluate the **beat-tracking performances**, we have used the F-measure proposed by Dixon [5] and the Gaussian error function proposed by Cemgil [2][11].

4.3 Configurations

We first compare various definitions of the set of tempo assumptions $\{B\}$.

Known Tempo (Q=1): $\{B\}$ is the annotated tempo. In this case, $Q = 1$ and the complex-search-code of u is correct. We therefore test the upper bound of the performances of our system.

Unknown Tempo (Q=5): $\{B\}$ is the reduced set of $Q = 5$ tempo assumptions defined by the tempo-estimation front-end.

Unknown Tempo (Q=321): $\{B\}$ is the whole set of $Q = 321$ tempo assumptions ranging from 60 to 220 bpm) without using any front-end.

We then compare the distances used to perform the search:

dE: the Euclidean distance is used to perform the rough search before the Complex distance.

dC: the One-minus-cosine distance is used instead.

dCP: the Complex distance is used to perform the whole search. Because of computation time, not all configurations are tested with this distance.

For comparison, we indicate in the "**IB**" rows the performances obtained using a dedicated tempo/ beat-tracking estimation system[12].

We finally compare the results obtained at the

Frame level: the target is t_i,

Item level: the target is u hence all the frames t_i belonging to u. For the results at the item level, we have used a **late-fusion integration** method, i.e. the method is applied for all the frames t_i of u and a decision is taken from the whole set of frames of u. The average number of frames t_i for an item u is 15. For deciding on the item class, we have used a majority voting method among the frame's classes. For deciding on the item tempo, we have used the median value over the frame's tempi. The "late-fusion integration" cannot be applied to the beat-tracking method. For beat-tracking, we therefore only present the results at the frame level.

In all cases, we have used a Leave-One-Out evaluation method, i.e. we test in turn each frame t_i of each item u as a target, and remove each time all the frames belonging to this u from the NN database. Therefore no frames that belong to the target item are used in the NN database.

4.4 Results and discussion

The results are indicated in Table 3. The gray areas represent configurations for which the estimation is not applicable. The top part of the table indicates the results considering all frames. In order to have a better understanding of the errors, we present in the bottom part of the table ("Only correct Classes"), the results obtained considering only the frames/ items which have been correctly classified with the corresponding configurations. Since the subset of frames/ items correctly classified is different for each configuration, we indicate for each one the value obtained with IB on the corresponding subset.

4.4.1 Using annotated tempo

When b_q is the annotated tempo ("Known tempo" columns) the best results for class accuracy are obtained using the One-minus-cosine distance: 82.6% at the frame and 93.1% at the item-level. The fact that the One-minus-cosine distance provides better results than the Euclidean distance is in agreement with previous studies [10]. Surprisingly, the exhaustive search using the Complex distance does not lead to the best results for class recognition. In the opposite, the best beat-tracking results are

[11] We have used the implementation of these criteria as provided by M. Davies in the evalbeat toolbox http://www.elec.qmul.ac.uk/ digitalmusic/downloads/ beateval/ beateval.zip.

[12] We have use the ircambeat software [27] [30]. This tempo/ beat-tracking system has been positively evaluated in the MIREX-09 and MIREX-10 contests.

#Frame: 10470 #File: 698		Known Tempo		Unknown (Q=5)		Unknown (Q=321)	
		Frame	Item	Frame	Item	Frame	Item
All							
Class Acc.	dE	82	92,1	52,9	58,3	54,6	61,9
	dC	**82,6**	**93,1**	**59,1**	**65,8**	**56,8**	**63,2**
	dCP	75,4	84,8	47,7	52,1	nc	nc
Tempo 4%	dE			60,4	61,5	62,6	65
	dC			**66,8**	**67,6**	**64,9**	**66,9**
	dCP			56,6	58,6	nc	nc
	IB			65,3	65,5	65,3	65,5
Beat Fmeas/Cemgil	dE	74,4 / 65,6		69,9 / 60,9		66.7 / 57,1	
	dC	74.7 / 65,9		71,6 / 62,5		68,2 / 58,5	
	dCP	**79,0 / 70,1**		**71,8 / 62,7**		nc	
	IB	78,4 / 71		78.4 / 71		78,4 / 71,0	
Only correct Classes							
Tempo (filter class)	dE			98,2	97,5	95,6	96,1
	IB			56,9	56	58.4	58,1
	dC			97,9	95,9	96,9	97,7
	IB			60,3	59,5	60,2	59,6
	dCP			98,2	97,8	nc	nc
	IB			60,3	59,5	nc	nc
Beat (filter class)	dE	76,4 / 67,8		83,4 / 74,4		79,6 / 70,2	
	IB	79 / 71.8		79,6 / 72,9		79,6 / 73,0	
	dC	76.7 / 68		82,0 / 72,8		**81,3 / 71.6**	
	IB	79.1 / 71.9		79,6 / 72,7		80,4 / 73,7	
	dCP	**81,6 / 73,1**		**88,2 / 79,3**			
	IB	79,1 / 72,2		79,4 / 72,9			

Table 3: Performance measures for classification, tempo estimation and beat-tracking in the case of known tempo, unknown tempo with $Q = 5$ tempo assumptions and with $Q = 321$; using Euclidean (dE), One-minus-cosine (dC), or solely Complex distance (dCP) with a decision at the Frame-level and Item-level decision. Comparison with the results obtained using a dedicated tempo/beat-tracking estimation system (IB). We denote by "nc" the "non-computed" values because of too high computation time.

obtained using the Complex distance: 79.0/70.1 which is above the results obtained using IB (78.4 and 71). It means that, when the correct tempo is used for creating $X_l(u, b, t_i)$, the beat-tracking performances of the "copy and scale" algorithm can be higher than the ones obtained with a state of the art dedicated beat-tracking algorithm. For the subset of correctly classified frames/ items, we observe the same ranking of the methods.

4.4.2 Using the reduced set of tempo assumptions

Using the reduced set of tempo assumptions ("Unknown Q=5" columns) the best class accuracy and tempo estimation are again obtained using the One-minus-cosine distance. It should be noted that the tempo estimations obtained (66.8% and 67.6%) are above the ones obtained with the dedicated algorithm IB. Again, while the Complex distance does not lead to the highest class accuracy or tempo estimation, it does again lead to the best results for beat-tracking: 71.8/ 62.7. This could indicate that a correct beat-tracking can be obtained by "copying and scaling" markers from items of different rhythm class and tempo. The beat-tracking performances are however 7% lower than with IB. For the subset of correctly classified frames/ items, the situation is different. All configurations of the "copy and scale" algorithms outperform the IB algorithm. The best results are obtained using the Complex distance both for tempo (98.2% and 97.8%) and beat-tracking (88.2/ 79.3). It means that, if the class is correctly identified, the proposed approach succeeds to estimate the correct tempo in 98% of the cases. This very high recognition rate has to be compared with the one obtained on the same frames/ items by the dedicated tempo-estimation algorithm (IB) which is much lower (around 60%). It should be noted that, because the tempo estimation performance of IB one these frames/items is lower than on the whole set, these frames/items do not appear to be the easiest ones.

4.4.3 Using the whole set of tempo assumptions

We now test our "copy and scale" method without using any front-end ("Unknown Q=321" columns). Because the computation time of the Complex distance is too high, we only indicate the results obtained using the Euclidean and One-minus-cosine distance. The best results are again obtained using the One-minus-cosine distance. The tempo estimations obtained with it (64.9% and 66.9%) are equivalent to the ones obtained with the dedicated algorithm IB. However the beat-tracking performances obtained are 10% lower than the ones of IB. Considering only the subset of correctly classified frames/ items, the situation is again different. All configurations of the "copy and scale" algorithms outperform the IB algorithm. The best results are obtained using the One-minus-cosine distance with tempo estimation of 96.9% and 97.7% and beat-tracking performances of 81.3/ 71.6. We therefore think that improving the classification part of our approach could lead to a very good tempo estimation and beat-tracking algorithm.

4.5 Comparison with previous results

Concerning **beat-tracking**, there is no previously published results on the "ballroom dancer" test-set (the annotation into beat positions has been made especially for the present paper). This was the reason for running the IB algorithm on it.

Concerning **tempo estimation**, previous published results are in the ISMIR-2004 tempo induction contest [16]: 63.2% (excluding octave estimation) and in our paper [27]: 68.7%. The results obtained here (67.6% with $Q = 5$ and 66.9% with $Q = 321$) can therefore be considered as nearly equivalent to the ones obtained by dedicated signal processing algorithms.

Concerning **classification** into rhythm classes, [6] obtained 85.7% track-based classification, we obtained 88% in [29] with an AdaBoost classifier. Results obtained here (65.8% with $Q = 5$ and 63.2% with $Q = 321$) with a Nearest Neighbor approach are therefore largely lower than the ones obtained in [29].

Considering that the classification part of our system can be improved (up to 88% using [29]) and considering that, for the part of correctly classified items our system reached 98% correct tempo estimation, one could therefore potentially reach a 87% correct tempo estimation (98% times 88%). The same is true for beat-tracking performances.

5 Conclusion and future works

In this paper, we proposed a new "copy and scale" method for estimating M.I.R. time-localized parameters and applied it to the problem of beat-tracking, tempo-estimation and classification into rhythm classes. In this method, time-localization is obtained by the use of a code in the complex domain, derived from the complex spectrum of an onset-energy-function. A simple Nearest Neighbor algorithm with a distance in the complex domain is then used to find the closest item of a pre-annotated database. Using this direct approach on the "ballroom dancer" test-set, a classification accuracy of 65.8% (63.2% without using a tempo prior front-end), a tempo precision of 67.6% (66.9% without) and a beat-tracking precision (F-measure) of 71.8% (68.2% without) are obtained. Analysis of the results shows that considering only the correctly classified frames leads to 98% tempo precision and 88% beat-tracking precision.

The results presented here should only be considered as a proof of concept of our method and testing our method using other test-sets with a wider diversity of rhythms needs to be done. However, considering the current results and the fact that the performances of each part of the proposed approach can easily be improved, we believe this method is promising. Potential improvements concern - the use of more sophisticated machine learning methods for rhythm classification as we did in [29], - the use of K-NN-regression for tempo estimation as proposed by [9], - making the beat-tracking performed at the frame-level benefits from the late-fusion integration performed at the item-level (tempo and class estimation benefits from the late-fusion integration at the item-level), - the introduction of temporal continuity constraints in the tempo, class and beat-marking decision (as we did in [28] using HMM to constrain tempo and class variations over time). Future works will also concentrate on the reduction of the search time in the NN database when using the Complex distance.

Acknowledgments

This work was partly supported by the "Quaero" Program funded by Oseo French State agency for innovation.

References

[1] Cemgil, A., and Kapen, B. Monte carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research 18* (2003), 45–81.

[2] Cemgil, A., Kappen, B., Desain, P., and Honing, H. On tempo tracking: Tempogram representation and kalman filtering. *Journal of New Music Research 29*, 4 (2001), 259–273.

[3] Davies, M., and Plumbley, M. Context-dependent beat tracking of musical audio. *IEEE Trans. on Audio, Speech and Language Processing 15*, 3 (2007), 1009–1020.

[4] Dixon, S. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research 30*, 1 (2001), 39–58.

[5] Dixon, S. Evaluation of audio beat tracking system beatroot. *Journal of New Music Research 36*, 1 (2007), 39–51.

[6] Dixon, S., Gouyon, F., and Widmer, G. Towards characterisation of music via rhythmic patterns. In *Proc. of ISMIR* (Barcelona, Spain, 2004), pp. 509–516.

[7] Eck, D., and Casagrande, N. Finding meter in music using an autocorrelation phase matrix and shannon entropy. In *Proc. of ISMIR* (London, UK, 2005).

[8] Ellis, D. Beat tracking by dynamic programming. *Journal of New Music Research 6*, Special Issue on Beat and Tempo Extraction (2007), 51–60.

[9] Eronen, A., and Klapuri, A. Music tempo estimation with k-nn regression. *IEEE Trans. on Audio, Speech and Language Processing 18*, 1 (2010), 50–57.

[10] Foote, J., and Uchihashi, S. The beat spectrum: A new approach to rhythm analysis. In *Proc. of ICME (IEEE Int. Conf. on Multimedia and Expo)* (2001), pp. 1088–1091.

[11] Goto, M. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research 30*, 2 (2001), 159–171.

[12] Gouyon, F., and Dixon, S. A review of rhythm description systems. *Computer Music Journal 29*, 1 (2005), 34–54.

[13] Gouyon, F., Dixon, S., Pampalk, E., and Widmer, G. Evaluating rhythmic descriptors for musical genre classification. In *Proc. of AES 25th Int. Conf. on Metadata for Audio* (London, UK, 2004), pp. 196–204.

[14] Gouyon, F., Dixon, S., and Widmer, G. Evaluating low-level features for beat classification and tracking. In *Proc. of IEEE ICASSP* (Honolulu, Hawaii, USA, 2007).

[15] Gouyon, F., Herrera, P., and Cano, P. Pulse-dependent analyses of percussive music. In *Proc. of AES 22nd Int. Conf. on Virtual, Synthetic and Entertainment Audio* (Espoo, Finland, 2002), pp. 396–401.

[16] Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., and Cano, P. An experimental comparison of audio tempo induction algorithms. *IEEE Trans. on Audio, Speech and Language Processing 14*, 5 (2006), 1832–1844.

[17] Grosche, P., and Muller, M. A mid-level representation for capturing dominant tempo and pulse information in music recordings. In *Proc. of ISMIR* (Kobe, Japan, 2009).

[18] Hainsworth, S., and Macleod, M. Beat tracking with particle filtering algorithms. In *Proc. of IEEE WASPAA* (New Paltz, NY, 2003).

[19] Holzapfel, A., and Stylianou, Y. Rhythmic similarity of music based on dynamic periodicity warping. In *Proc. of IEEE ICASSP* (Las Vegas, USA, 2008).

[20] Holzapfel, A., and Stylianou, Y. A scale tranform based method for rhythmic similarity of music. In *Proc. of IEEE ICASSP* (Taipei, Taiwan, 2009).

[21] Jehan, T. *Creating Music by Listening.* Phd thesis, Massachusetts Institute of Technology., 2005.

[22] Klapuri, A. Sound onset detection by applying psychoacoustic knowledge. In *Proc. of IEEE ICASSP* (Phoenix, Arizona, USA, 1999), pp. 3089–3092.

[23] Klapuri, A., Eronen, A., and Astola, J. Analysis of the meter of acoustic musical signals. *IEEE Trans. on Audio, Speech and Language Processing 14*, 1 (2006), 342–355.

[24] Laroche, J. Efficient tempo and beat tracking in audio recordings. *Journal of Audio Engineering Society 51*, 4 (2003), 226–233.

[25] MIREX. Audio beat tracking contest, 2009.

[26] Paulus, J., and Klapuri, A. Measuring the similarity of rhythmic patterns. In *Proc. of ISMIR* (Paris, France, 2002), pp. 150–156.

[27] Peeters, G. Template-based estimation of time-varying tempo. *EURASIP Journal on Applied Signal Processing 2007*, 1 (2007), 158–158. doi:10.1155/2007/67215.

[28] Peeters, G. Template-based estimation of tempo: using unsupervised or supervised learning to create better spectral templates. In *Proc. of DAFX* (Graz, Austria, 2010).

[29] Peeters, G. Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal. *IEEE. Trans. on Audio, Speech and Language Processing* (in press 2011).

[30] Peeters, G., and Papadopoulos, H. Simultaneous beat and downbeat-tracking using a probabilistic framework: theory and large-scale evaluation. *IEEE. Trans. on Audio, Speech and Language Processing* (in press 2011).

[31] Peeters, G., and Rodet, X. Non-stationary analysis/synthesis using spectrum peak shape distortion, phase and reassignement. In *Proc. of ICSPAT (Int. Conf. on Signal Processing Applications and Technology)* (Orlando, Florida, USA, 1999), D. World, Ed., DSP World.

[32] Scheirer, E. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America 103*, 1 (1998), 588–601.

[33] Tzanetakis, G., and Cook, P. Musical genre classification of audio signals. *IEEE Trans. on Speech and Audio Processing 10*, 5 (2002), 293–302.

[34] Wright, M., Schloss, W., and Tzanetakis, G. Analyzing afro-cuban rhythms using rotation-aware clave template matching with dynamic programming. In *Proc. of ISMIR* (Philadelphia, PA, USA, 2008), pp. 647–652.