A causal algorithm for beat-tracking

Benoit Meudic
Ircam - Centre Pompidou
1, place Igor Stravinsky, 75004 Paris, France
meudic@ircam.fr

ABSTRACT

This paper presents a system which can perform automatic beat-tracking in real-time from performed polyphonic music in symbolic format. No voice information is needed. The music can be as well a performance recording containing tempo changes and irregularities, or a score transcription. The needed input format is a midi-like sequence containing at least the onset, duration, and pitch features. In certain cases, the only onsets could also provide good results for some rhythmic structures. The chosen approach is quite similar to the one proposed in (Dixon 2001), however Dixon's method can not be implemented in real time. Another difference is that we make an intensive use of markings, in order to detect salient rhythmic events, which offers the possibility to adapt the behavior of the algorithm according to the type of music which is analyzed. The system was tested on Ragtime pieces and on two Beatles songs with satisfying results.

1. INTRODUCTION

During the last few years, automatic beat-tracking, which aims to extract a beat from a music file, has been a topic of active research. In the context of Music Information retrieval (MIR), beat-tracking constitutes an important area of research, because rhythm is one of the main features needed for applications such as query by humming, music abstracting or pattern analysis. In this article, we will focus on beat-tracking in real time from performed polyphonic music. The method we propose will be compared to (Dixon 2001) because both methods use a similar approach.

Several definitions of beat can be found in the literature, with even different terms for similar notions. In this paper, we will focus on the extraction of one regularly recurring stimuli in a musical sequence, which will be called *beat*, and which is expected to be simple multiple of the beat represented in the score (if available) or simple multiple of the rhythmic regularity one can perceive when listening to the music. Then, if needed, other metric levels could be extracted with algorithms such as the one proposed in (Meudic 2002). When compared to a periodic signal, the beat can be described by its period value (a duration, often expressed in milliseconds) and its phase values. In the article, we will employ the word phase values to reference the different onset times at which the beat occurs in a sequence.

When listening to the music, we often mark a beat quite easily, even not conscientiously. However, by trying to extract it automatically from a music sequence, we are confronted to the machine, which is not sensitive to music. We think that a good way to approach this issue is to model the phenomenon of expectancy, which could be linked to the processes employed by human listeners in their understanding of rhythm. Indeed, when listening to music, we often mark the beat, and in order to mark it correctly, we have to anticipate it, otherwise, there would be a difference between the beat time occurrence and our marking of it, which would come from the time we need for reacting to the beat occurrence. Thus, we expect the beat to occur. Sometimes, even after the music stops, we continue marking the beat a few seconds. We believe that the idea to get close to the models of human perception of rhythm are good ways to provide interesting automatic algorithms.

Following this idea, (Desain et al 1990) propose to use the auto-correlation measure to track the beat along a musical sequence. However, auto-correlation hardly provides information on the phase values (time occurrences) of the beats.

(Large et al 1994) model rhythmic expectancy by using oscillators. In this case, the principle is not to extract the beat directly from the events sequence, but from an intermediate model (the oscillators) which would be sensitive to the periods of the sequence. The method has the advantage to be implemented in real time. A drawback is that no other information than inter-onsets is taken into account, whereas rhythm can sometimes be influenced by other features such as dynamics or pitches.

Sometimes, the methods integrate a kind of musical knowledge. (Cemgil et al 2000) propose to use a stochastic dynamic system in a Bayesian framework. The tempo is modeled as a hidden state variable of the system and is estimated by a Kalman filter. The method has the advantage to be real time implemented, except that it requires a previous learning stage on the data during which Kalman filters are trained.

(Dixon 2001) propose a non real-time multiple agent model which induces several beats and then tries to propagate them in the musical sequence. The method uses musical knowledge (Dixon et al 2000), but it does not appear as a drawback because it is general enough to be applied to various styles of music. This approach is particularly interesting, and will be further described in the following chapter.

Our system is based on an approach quite similar to the one proposed by Dixon, but it contains several improvement : the algorithm is causal, parametrisable, and sensitive to complex tempo changes. After having described the two systems, theses improvements will be more detailed.

2. DIXON'S ALGORITHM

The model proposed by Dixon can be divided into three steps:

- (1) First, a list of possible beats is induced from the beginning of the music sequence
- (2) Then, the algorithm tries to propagate the beats along the analyzed sequence, that is to say it chooses the events in the sequence which could correspond to the beats occurrences.

- (3) Lastly, the list of the beats which have been propagated is sorted according to several criteria (among which a kind of musical knowledge is used) in order to select the 'best beat'.

The beat induction step (1) analyses the inter-onsets of the events contained in the first few seconds of the beginning of the sequence. Clusters of inter-onsets are created, and for each cluster, a beat value (what we called the period of the beat) is extracted. Then, for each extracted beat value, the algorithm generates as many agents as there are events in the windowed sequence, each agent corresponding to a given beat value and to the given temporal occurrence of an event (what we called the phase value of the beat).

The second step (2) consists in propagating the agents in the sequence. For each event, for each agent, the algorithm states if the event can correspond to the expected time occurrence of the agent or not. If the event corresponds, a weight is added to the agent's score (see step (3)). If there is ambiguity, the algorithm duplicates the agent to answer both yes and no. Assuming that several events occur in the expected occurrence area of an agent, one can guess that the duplication of agents will greatly increase. In order to reduce this number, the algorithm removes agents which are stated as similar.

The third step (3) consists, after having analyzed the entire sequence with (1) and (2), in selecting the agent which corresponds the best to the expected beat. For this, (Dixon et al 2000) use what they call "musical knowledge" : each event of the sequence is weighted proportionally to the values of its features such as duration, dynamics and pitches. Then, for each agent, a score is established by considering the weights of the events corresponding to the temporal occurrences of the agent. The weights are pondered by a value proportional to the time distance between the expected time positions of the agent and the time positions of the chosen events. The agent with the highest score is designed as the most convenient.

3. THE CAUSAL BEAT-TRACKING ALGORITHM

Our beat-tracking algorithm is composed of three main interactive and complementary functions (f1), (f2), and (f3) which are processed step by step on each event of the music sequence. During the process, the events of the sequence are marked (see 3.1) so that the salient events can influence the choices made by the three functions. A list of possible beats (l.p.b) is regularly updated according to the output of the three functions (see 3.2). Each beat is described by its period and by the list of its phase values in the sequence already analyzed. Each beat is expected to occur periodically (see Figure 1). The next expected phase value of a beat is calculated by adding its current period to its last phase value. The function (f1) induces possible beats (see 3.3) and (f2) confirms or cancels their expected temporal downbeats in the music sequence (see 3.4), while (f3) outputs a beat from the (l.p.b) which should correspond to the 'most relevant one' (see 3.5). Contrary to Dixon's algorithm, the analysis is causal as none of the functions use the informations of the events following the one currently analyzed.

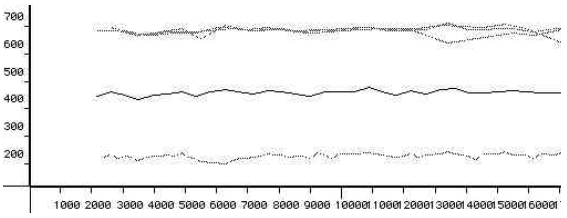

Figure 1. The evolution in time of the list of possible beats (l.p.b). The period (in ms) is represented vertically, and the horizontal axis is time. One can distinguish three main different periods. The irregularities are due to tempo variations in the performance. The upper line is the superposition of two similar period with different phase.

3.1 The markings

One basic idea which stands behind the notion of marking is that it provides a hierarchy between events belonging to a sequence. Several algorithms, theories and methods have used what could be called markings but did not explicit this notion as a specific concept (a theory which proposes a language to manipulate and describe the markings can be found in [Lusson 86]).

Considering a property which can be applied to a sequence of events (for instance the property "duration" for a sequence of notes), what we call a marking is the association between the sequence of events and the sequence of weights resulting from the property applied to each of the events of the sequence.

The link between the markings and the beat-tracking issue is that we assume that rhythmically important events are related to beat temporal positions.

In our algorithm, this relationship holds between the two parameters of a beat :

- its period : a beat period is more likely to correspond to the temporal distance between two rhythmically important events than to the distance between two arbitrary events (see 3.3 for application).

- its phase value : a beat is more likely to occur on rhythmically important events (see 3.4 for application).

To establish the markings, we consider three musical parameters (available in the midi format) : pitches, durations and onsets.

Four markings have been made :

· Marking1 uses the onset information to mark repetitions : are weighted with the weight 1 the events which ioi (inter-onset) with the event before has been repeated at least one time consecutively. The notion of repetition is not absolute for performed music, so a threshold is considered.

For instance :

the onset sequence (in ms) :

(0 10 50 90 110)

is marked (? 0 0 1 0)

· Marking2 also uses the onset information to mark long inter-onsets : are marked with the weight x the events which ioi whith the next event is greater than the x consecutive past ioi. There again, a threshold is considered.

For instance :

the sequence (0 10 50 100 110)
is marked (0 1 2 0 ?)

- Marking3 uses the pitch information to mark the note densities : are marked with the weight 1 the events which have more than one note. Notes with slight onset differences are considered as being part of the same event.

  For instance :

  the sequence ((C) (C E G) (D) (G B D) (C))
  is marked (0 1 0 1 0)

- Marking4 uses the onset and duration features to mark coverings : are marked with the weight x the events which occur at the end of the duration of a previous event which covers x>1 events.

  For instance :

  the sequence (onset, duration) :
  ((0,90) (10,40) (50,40) (90,20) (110,?))
  is marked
  (0 0 0 2 0)

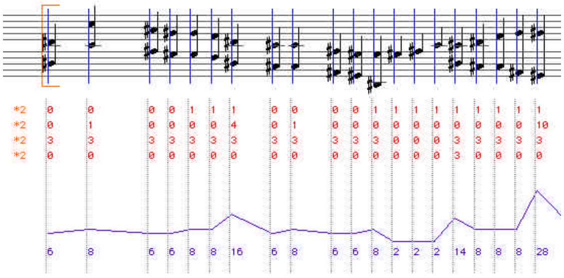

Figure 2. The four markings applied to a musical sequence and the graph of their linear combination

We believe that according to the different combinations of thoses markings, different musical contexts can be considered. For the moment, the final weighted sequence is computed by doing a linear combination of the different markings (see Figure 2). Each marking is causal, so the weights of the events can be computed at each step of the process without knowing the events which follow.

3.2 The processing in time

At the beginning of the process, the list of the possible beats (l.p.b) is empty. Then, the sequence is analyzed event after event. For each event, (f1), (f2) and (f3) are applied. Before analyzing the next event, the l.p.b is updated according to the output of (f1) and (f2).

According to the cases, the update consists in :

- Adding new beats to the l.p.b according to the output of (f1). In order to control this step, one parameter, 'beat-max', controls the maximum total number of beats which can be part of the l.p.b. If the maximum number of beats is reached, the new beats won't be taken into account.

- Deleting from the l.p.b some beats which have not been updated since a given time. A parameter, 'context-beat', determines the number of times that the beats can repeat without being updated.

- Updating the beats in the l.p.b which are found to occur on the current event (output of (f2)). Both period, phase value and weight of the beats are updated. One can imagine three different situations :

- The current event exactly corresponds to an expected position of the pulse. In this case, the onset of the event is added to the list of phase values of the l.p.b, and the frequency of the pulse is not modified.

- The current event approximatively corresponds to an expected position of the beat (which is determined by (f2)). In this case, the above operations are still performed, but the expected frequency of the pulse is averaged with the found period.

- The current event corresponds to an expected position of the beat, but another event had already been chosen for this expected position. If the current event is less convenient than the already chosen one (this is determined by (f2), see 3.4), nothing happens. Otherwise, the last event which was determined to correspond to the expected occurrence of the pulse is deleted from the l.p.b to the profit of the current event.

3.3 Function (f1) : the hypothesis of new beats

This function analyses a marked sequence of events of length 'length-context' in order to extract new possible beats. We assume that the onset of the last event of the sequence, which is the current event of the analysis, corresponds to a phase value of a new beat (this assumption is determined by (f2)). Then, we select all the inter-onsets between this event and the other events of the sequence. Each inter-onset is said to determine a new beat whose first phase value is the phase value of the current event. Then, the function outputs the new beats.

3.4 Function (f2) : the propagation of the phases values

This function analyses a marked sequence of events of length 'tolerance-window' in order to determine if the onset of the last event of the sequence, which is the current event of the analysis, corresponds to a possible phase value or not.

Two different cases can be considered :

- the onset corresponds to a phase value of a new beat which is not in the l.p.b.

To validate this hypothesis, the function just checks if the weight associated to the onset is higher than the weight of its neighbors. If yes, (f1) is called.

- the onset corresponds to the expected phase value of a beat which is in the l.p.b. (see Figure 3)

To validate this hypothesis, each expected phase position of each beat of the l.p.b is considered. The value of 'tolerance-window' is proportional to the current period of the considered beat (we have chosen the ratio 0.2). The highest the period, the highest the length of the window. A new marking is introduced such as the nearest events of the expected phase position are weighted by the greatest value and inverse for the other ones. Then, the values of the new marking are added to the weights of the marked sequence. The last onset of the sequence is said to correspond to the expected phase position if its new weight is higher than the weight of the event already chosen for this expected position. If there was no event chosen for the position, the last onset is automatically considered as convenient.

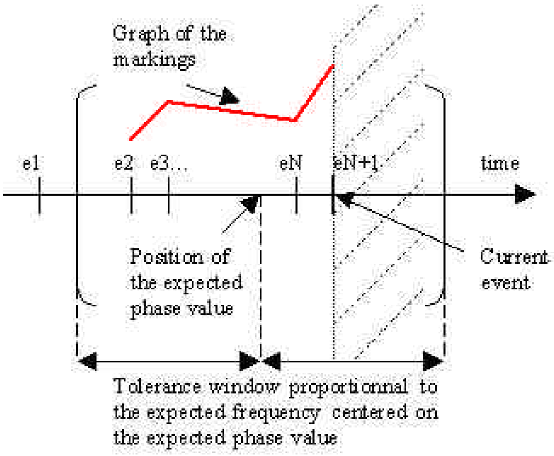
Graph of the markings
e1
e2
e3...
eN
eN+1
time
Position of the expected phase value
Current event
Tolerance window proportionnal to the expected frequency centered on the expected phase value

Figure 3. The propagation of the phase values. The event eN+1 is confirmed by (f2) as a possible phase value because it is included in the tolerance window and its weight is higher than the weight of the last confirmed event.

3.5 Function (f3) : the extraction of the best beat

At each step, (f3) analyses the l.p.b in order to extract the 'best beat'. The l.p.b memorizes for each occurrence of each beat the corresponding period, phase value and weight.

We define the 'best beats' as being the ones which would be represented in a score or the ones we would be perceived when listening to the sequence.

To help us understand how such beats could be extracted from the l.p.b, the l.p.b can be visualized in a graph (see Figure 1). The graph greatly helps us to analyze the results, as we can simultaneously visualize the various evolutions in time of the possible beats.

With little experience, "good" beats can rapidly be visualized. Thus, we have to define why, in the graph, some beats appear to us as being the "good" pulses.

In order to sort the pulses, several criteria could be tested :

- (1) The number of confirmed phase positions of the beat in the sequence divided by the mean of the period of the beat. The aim is to erase the beats which are confirmed only on a small part of the analysed sequence.
- (2) The weight of the beat (sum of each weight for each confirmed phase position of the beat) divided by the mean of the period of the beat. The aim is to detect the beat which falls on the more weighted events of the sequence.
- (3) The "regularity" in the evolution of the graphic profile of the beat. Indeed, we could assume that very chaotic profiles can not correspond to the expected beat.
- (4) The number of the multiples and sub-multiples periods of the beat contained in the list. Indeed, one can think that several beats which are in harmonic relation (that is to say which phase values are similar and which periods have simple proportionnal relations) are more likely to correspond to the expected beat than an isolated beat.
- (5) The distance of the beat from the value "60 quarter notes per minutes" which is the tempo value the most often found in most of the scores.

After having tried different combinations of the criteria, it appeared that the rules **(1)** and **(3)** were the most relevant to select the 'best' beat, and thus those criteria were chosen for our algorithm.

4. COMPARISON OF THE TWO ALGORITHMS

The main characteristic of the causal algorithm, when compared to Dixon's, is that it can be implemented in real-time, whereas Dixon's algorithm would require deep changes before being implemented in real-time. In our system, the three consecutive steps (induction, then propagation and then final beat extraction) are processed at the same time for each event of the sequence. The simultaneous processing of the induction and the propagation phase values solves the issue of the free introductions which was raised by Dixon : instead of inducing beats from the only first seconds of the beginning, which sometimes contain free rhythm, we induce beats during all the processing of the sequence. If wrong beats were induced at the beginning and if they are not confirmed by the propagation step, they would be erased while new beats would be induced.

Moreover, the markings are used not only in the final beat extraction step, but also in the two other steps : Concerning the beat induction, the markings filter the events so that the only most weighted ones are taken as possible positions for the phase value of new beats (whereas Dixon considers all the positions of all the events contained in a given window). Concerning the beat propagation, the algorithm selects the event the most weighted in the tolerance window whereas Dixon duplicates the agents when ambiguity arises. Doing this, we dramatically reduce the number of possible beats, which makes our algorithm faster. In our system, markings are crucial for performing beat-tracking with success (they could be compared to the cane of a blind man). If the different steps (induction, propagation, extraction) of our algorithm can not be modified, the markings can be seen as parameters and thus allow any user to control the behavior of the algorithm according to its own model of what are rhythmical salient events. For instance, in our system, Marking1 can compute the salience of drum sounds (a drum sound is salient because it is repeated) which was said to be lacking in Dixon's algorithm.

Another characteristic stems from the analysis of the tempo variations : in Dixon's algorithm, when the occurrence of an event corresponds to the expected position of a beat, it is definitively accepted as a new occurrence of the beat. In our algorithm, the event is accepted, but if a following event positioned in the tolerance window reveals to be more weighted, the accepted event is forgotten to the benefit of the newly considered one. Doing this, on the contrary to Dixon, we can detect tempo variations even if bad events occur at the expected position of a beat.

Finally, our method can also analyze the harmonic relations between the different induced beats, which allow to group them in clusters. This could be used for instance to filter the beats along the processing of the sequence in such a way that only relevant clusters (clusters containing more than n beats) would be kept in the l.p.b. However, we didn't implement yet this functionality in our algorithm.

5. TESTS AND RESULTS

Our beat-tracking algorithm has been implemented in Open-Music [agon 98]. The input data are Midi-files. The parameters which we consider are : onsets, pitches and durations. No information on tracks or voices is used. In a pre-processing step, the midifiles are filtered in order to regularize micro-deviations of onsets : onsets which are distant of few milliseconds each others (as this is often the case in performed music) are grouped into chords, ie each event of the group is given the same onset value.

5.1 Tests on ragtimes containing artificial tempo variations

We have tested our algorithm on four different ragtime pieces transcribed from a score on which different artificial tempo stretching have been made, in order to simulate a kind of performance

(the stretched Midi-files are available at http://www.ircam.fr/equipes/repmus/meudic).

The tempo stretchings were successive dilatation and compression of the onsets and the durations. Five different stretching have been made, from the less to the most complex.

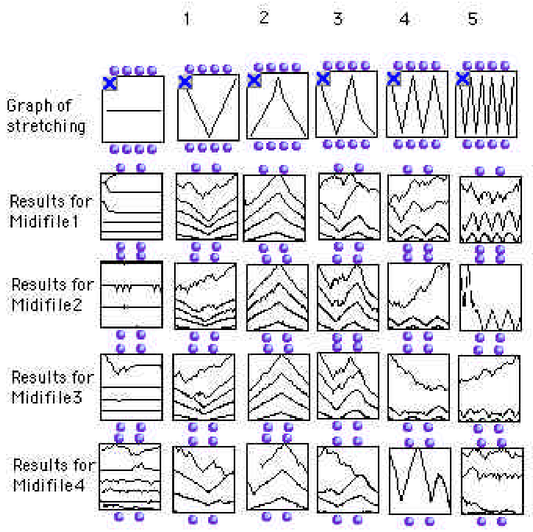

Figure 4. Different evolutions in time of the l.p.b.

Each box in Figure 4 show the evolution graph along time of the list of possible beats periods (expressed in milliseconds) extracted by the algorithm for each stretch applied to each Midi-file. One can notice that high periods have often an erratic behavior whereas small ones follow with more accuracy the curve of the stretching applied to the sequence. In all the cases except for the last column (corresponding to the most complex stretching), the algorithm automatically extracts a beat which follows with accuracy the tempo variations applied to the Midi-file. The erratic curves were never chosen by the algorithm.

5.2 Comparison with two other tempo trackers for two Beatles songs

We have tested our system on a dataset of 219 different performances of two beatles song, Michelle and Yesterday, which were collected for the testing of the beat tracking system reported in (Cemgil et al 2000). Dixon's beat tracking system has also been tested on this dataset (Dixon 2001b).

The same evaluation procedure was employed for the three models. It consists in rating the similarity of two sequences of beat times (the expected sequence and the sequence proposed by the algorithm) as a percentage.The procedure is fully described in (Cemgil et al 2000).

In order to compare the three systems, the "beat induction" step was not performed automatically, because the beat period is considered as a given input in the system of Cemgil.

Table 1. Average tracking performance and standard deviations by subject group, tempo condition, and average over all performances

	Yesterday	**Michelle**
By subject group		
Professional jazz	97 +/- 1	97 +/- 1.5
Amateur classical	95 +/- 2.5	96 +/- 2
Professional classical	92 +/- 3	88 +/- 5
By tempo condition		
Fast	95 +/- 2	93 +/- 5
Normal	94 +/- 3	94 +/- 4
Slow	94 +/- 3	93 +/- 5
Average	**94 +/- 3**	**93 +/- 5**

The results are reported in Table 1. The processing time for each song was about half a second which is very satisfying (Dixon reported between 2% and 10% of the length of the music, which would correspond to 1 to 5 seconds for the Beatles songs). The tracking performances are very similar to the ones reported by the two other authors. The results for the category Professional classical are a little worse than for the other categories, both for Yesterday and Michelle songs. This was also reported by the other authors. Future work could try to enhance the algorithms results for this particular style.

We think that such experiments are very valuable both for improving and comparing the algorithms. However, such midi databases of performances are rarely available, and thus the research community should make an effort to provide them.

6. CONCLUSION

We have presented an algorithm which extracts a beat in real-time from performed polyphonic symbolic music. The approach is similar to (Dixon 2001), but it is faster and can be implemented in real-time. This improvement stands in the fact that the three main functions are causal and are processed simultaneously on each event of the music sequence whereas they where processed in three different steps in Dixon's algorithm. Moreover, the use of the markings in the three functions reduces the complexity of the list of possible beats and makes the algorithm process faster. The examples showed that the algorithm could adapt to big tempo changes and that its performances in extracting the beat from the two Beatles songs where similar to the ones published in (Dixon 2001b) and (Cemgil et al 2000).

In our algorithm, the markings play a crucial role in both the choice of the periods and the choice of the phase values of the extracted beats and thus, one should think that the efficiency of the algorithm mainly stands in the choice of relevant markings. Many experiments that we have done but which are not presented in this article show that if rhythmically salient events are correctly detected, then the algorithm would successfully finds the beat. This can be seen as a positive point because the issue of beat-tracking would be transformed in another issue, maybe less complex, which is : "how to extract rhythmical salient events from a musical sequence" ?

We think that the modeling of different rhythmical styles with the help of different markings would be a great step forward in the analysis of rhythm by a machine. The idea of finding universal markings which would be adapted to any musical style is appealing, but maybe not realistic. Anyway, this issue should be studied deeply and will be part of our future work. Note that markings can also play a role in the extraction of other rhythmic component such as meter (Meudic 2002).

One issue which still remains is how to measure the performance of the algorithm. We think that the comparison of the output of the algorithm with a score is sometimes a nonsense. For instance, musicians can introduce rhythmic effects like groove which makes us feel that the beat occurs a little after or before the event on which it should occur. Note that this should not be confound with the use of rubato which introduces big deviations from the score, but which correspond to perceived beat deviations (this is another issue). The question is : should we extract the beat as represented in a score, or should we extract the beat as played by the musician? In other words, should we recompose the initial rhythmic score from the free performance or should we establish a rhythmic score of the performance? Most of the current beat-tracking algorithms (including ours) extract the beat as played in the performance, and thus results should be compared to the rhythmic score of the performance. The problem is how to get it, knowing that a simple matching between initial score and performance would not be convenient because the two are different.

7. ACKNOWLEDGEMENTS

This research was partly supported by the european project CUIDADO which aims at developing new content-based technologies for music information retrieval. I want to thank Cemgil, Kappen, Desain and Honing to have made available their Beatles data set. I also want to thank Gérard Assayag for his suggestions about this article.

8. REFERENCES

(Agon 1998)

"Openmusic, un langage visuel pour la composition musicale assistÉe par ordinateur."

Thesis, Ircam 1998

(Cemgil, Kappen, Desain, Honing, 2000)

"On tempo tracking : Tempogram representation and Kalman filtering"

ICMC 2000 p.352-355

(Desain - Siebe de Vos 1990)

"Auto-correlation and the study of Musical Expression"

(Dixon, Cambouropoulos, 2000)

"Beat tracking with musical knowledge."

ECAI, 2000

(Dixon 2001)

"Automatic Extraction of Tempo and Beat from Expressive Performances."

Journal of New Music Research, 30, 1, 2001, pp 39-58.

(Dixon 2001b)

"An empirical Comparison of tempo trackers"

8th Brazilian Symposium on Computer Music, 31 July - 3 August 2001, Fortaleza, Brazil.

(Large et al 1994)

"Resonnance and the Perception of musical meter"
Connection Science, 6 (1) 177-208

http://www.cis.upenn.edu/~large/publications.html

(Lusson 1986)

"Place d'une théorie générale du rythme parmi les théories analytiques contemporaines"

Analyse Musicale n°2, pp. 44-51, 1986.

(Meudic 2002)

"Automatic meter extraction from midifiles"

Jim 2002 - Marseille