A Learning-Based Quantization: Estimation of Onset Times in a Musical Score

Masatoshi Hamanaka* **Masataka Goto**,*****
Hideki Asoh*** **Nobuyuki Otsu***,***

***University of Tsukuba** ** "Information and Human Activity", PRESTO, JST
****National Institute of Advanced Industrial Science and Technology (AIST)**
Mbox 0604, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan
m.hamanaka@aist.go.jp

ABSTRACT

This paper describes a method for organizing onset times of musical notes performed along a jam-session accompaniment into the normalized (quantized) positions in a score. The purpose of this study is to align onset times of a session recording to quantized positions so the performance data can be stored in a reusable form. Unlike most previous beat-tracking-related methods focusing on predicting or estimating beat positions, our method deals with the problem of eliminating the onset-time deviations under the condition that the beat positions are given. To quantize polyphonic MIDI recordings of jam session, we propose a method that uses hidden Markov models for modeling onset-time transition and deviation. Its main advantage is that a player's performance is quantized using a model learned statistically from session recordings of that player. Experimental results show that our model performs better than the semi-automatic quantization in commercial sequencing software.

Keywords: Quantization, Hidden Markov model, Recognition of beat and rhythm, Statistical learning

1. INTRODUCTION

We have been constructing a jam session system [1] that allows a human player to play interactively with virtual players, each of which is imitating the musical reactions of a human player. Each virtual player determines its intentions by using a reaction model that has been acquired from a human player and then produces a performance by connecting short phrases selected from a database of phrases. Since this database was prepared by hand, the system could not imitate the player's characteristic phrases automatically.

Simply cutting out phrases at bar lines and pasting them does not work well (it creates unnatural performances) because the onset times of notes played by human players intentionally or unintentionally deviate from the 'normal' position of onset times in a score. Before cutting and pasting phrases, we need to use a quantization method that eliminates the deviation of onset times and aligns them to the normalized positions[1] in a score. A quantization method typical of commercial sequencing software requires the user to specify a fixed grid interval, or resolution, (e.g., eighth triplet or sixteenth note) to which onset times are aligned, and each onset time is aligned to the nearest grid. This method can therefore be used only when the rhythm structure within a beat is fixed and known (e.g., the beat contains eighth triplets or the beat contains sixteenth notes). When the rhythm structure changes frequently, as it does in a jam session, we need to change the grid interval adaptively.

Several quantization methods have been proposed, and one using a connectionist model [2] defines a *potential energy* that is stable if the ratio of a sum of onset time intervals to a sum of other intervals is an integer. It is not easily applicable to various performances, however, because the *potential energy* is fixed. Another quantization method for automatic transcription [3] and beat-tracking methods [4]-[8] focusing on predicting beat positions cannot be applied to our study directly, because their problems (estimation of beat position) are different from our problem (quantization of onset times performed along a fixed-tempo jam-session accompaniment[2]).

On the other hand, the results of a study [8] dealing with a problem similar to ours indicate that the continuous speech recognition framework using a hidden Markov model (HMM) provides a useful approach to estimating tempos and beats and to allocating bar lines. The method using that approach performs better than commercial sequencing software does, but it deals only with single-note performances and thus cannot be used to quantize the recording of a jam session that we deals with.

---

[1] Because in improvisation there is no score to follow, here *the normalized position in a score* means the position at which the player intended to play that note.

[2] Quantization of commercial sequencing software is not effective in this problem, because an onset time that has large deviation is aligned to an incorrect grid.

In this study, we propose a method that uses the promising approach proposed in the study [8] and makes it possible to quantize polyphonic MIDI recordings of jam session by using our own HMM-based model.

2. LEARNING-BASED QUANTIZATION

A human player, even when repeating a given phrase on a MIDI-equipped instrument, rarely produces exactly the same sequence of onset notes because the onset times deviate according to performer's actions and expressions. We can model the process generating the deviation by using a probabilistic model. Then the problem of quantization, which acquires a sequence of onset times that the player intended from a sequence of deviating onset times that the player actually performed, can be considered as an inverse problem. This inverse problem can be solved by using the inverse model derived from the model generating the deviation of onset times.

A model of onset-time transition and deviation

Let a sequence of intended (normalized) onset times be $\boldsymbol{\theta}$ and a sequence of performed onset times (with deviation) be $\boldsymbol{y}$. Then a model of generating the deviation of onset times can be expressed by a conditional probability $P(\boldsymbol{y}\,|\,\boldsymbol{\theta})$ (Figure 1). Using this conditional probability and the prior probability $P(\boldsymbol{\theta})$, the inverse model can be calculated as Eq. (1) according to the Bayes' theorem:

$$P(\boldsymbol{\theta}\,|\,\boldsymbol{y}) = \frac{P(\boldsymbol{y}\,|\,\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{y})}, \qquad (1)$$

where $P(\boldsymbol{\theta})$ represents how likely it is that a player plays the sequence of onset times $\boldsymbol{\theta}$. Because $P(\boldsymbol{y})$ is independent of $\boldsymbol{\theta}$, it can be ignored. Thus the solution to the inverse problem of determining optimal $\hat{\boldsymbol{\theta}}$ can be obtained by maximizing Eq. (1):

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P(\boldsymbol{\theta}\,|\,\boldsymbol{y}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P(\boldsymbol{y}\,|\,\boldsymbol{\theta})P(\boldsymbol{\theta}). \qquad (2)$$

Formulation of the hidden Markov models

$P(\boldsymbol{\theta})$ and $P(\boldsymbol{y}|\boldsymbol{\theta})$ can be formulated as a hidden Markov model (HMM), which is a probabilistic model that generates a transition sequence of hidden states as a Markov chain. Each hidden state in the state transition sequence then generates an observation value according to an observation probability.

Modeling of performance:

- ***Target in modeling***

  We model the onset time of a musical note (i.e. the start time of the note) and introduce a new model of distribution of onset times. While the duration-time-based model used in Ref. [8] is limited, our onset-time-based model is suitable for treating polyphonic performances, such as those including two-hand piano voicing and guitar arpeggio.

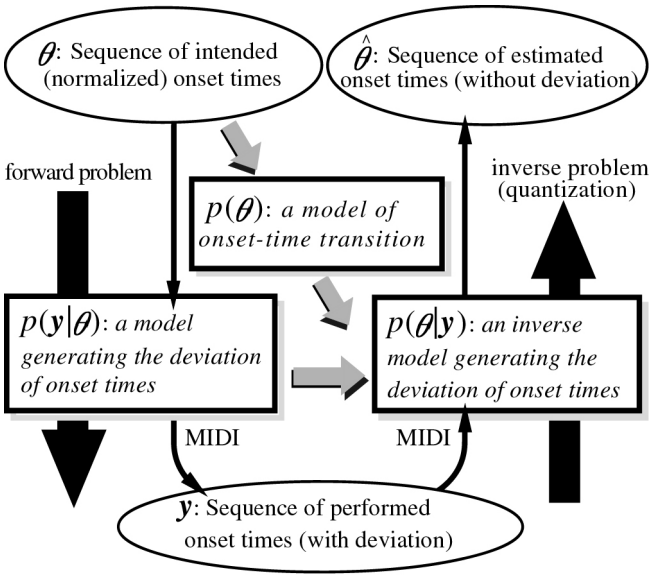

Figure 1: Forward model and inverse model in the quantization problem.

- ***Unit in modeling***

  We use a quarter note (beat) as the unit of each HMM: the temporal length corresponding to each HMM is a quarter note. The reason we use the quarter-note unit is to distinguish between eighth triplets and sixteenth notes within the scope of a quarter note. The three notes of eighth triplets are located on three equi-divided positions in a quarter note, while the four notes of the sixteenth notes are located on four equi-divided positions in a quarter note. An actual performance consisting of a sequence of quarter notes can be modeled and quantized by concatenating the quarter-note-length HMMs.

  This quarter-note modeling has the advantages of reducing calculation time and facilitating the preparation of the large data sets used for training the model.

- ***Unit of quantization***

  We introduce two different discrete temporal indices, *k* and *i*. The unit of *k* is a quantization unit to describe performed onset time and is 1/480 of a quarter note, which is often used in commercial sequencing software. The unit of *i* is a quantization unit to describe the intended onset time and is one-twelfth of a quarter note. It can describe both eighth triplets and sixteenth notes.

Quarter-note hidden Markov model: Figure 2 shows the HMM used in our study. We model a sequence of onset times within a quarter note (beat) by using the HMM. All the hidden states of the HMM correspond to possible positions of intended onset times, and an observed value that comes from a hidden state corresponds to a performed onset time with deviation. Onset times in a beat are quantized into 12 positions for hidden states, and into 480 positions for observation values. That is, each component of the HMM is interpreted as follows.

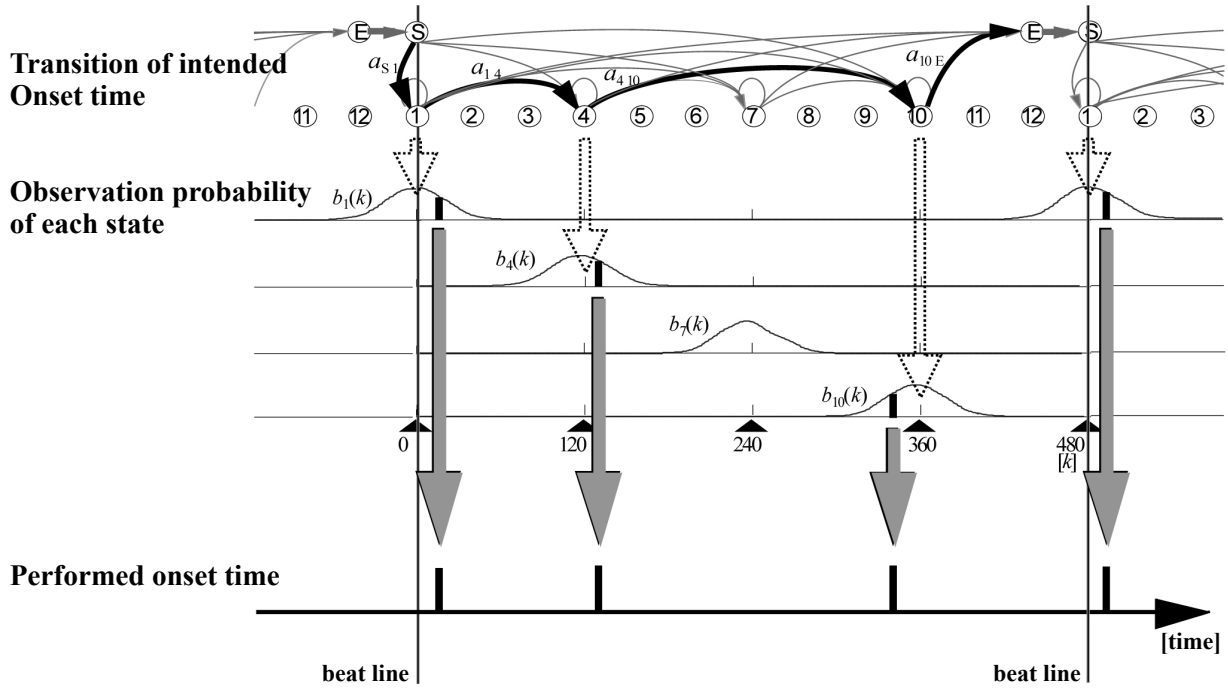

Figure 2. Overview of the quarter-note hidden Markov model

Hidden state *i*: intended onset time. (*i* =1, …, 12)
Observation *k*: performed onset time. (*k* = 1, …, 480)
Transition probability *aij*: probability that intended onset time *j* follows intended onset time *i*.
Observation probability *bi*(*k*): probability that performed onset time is *k* and intended onset time is *i*.
A state-transition sequence begins with a dummy state "Start" and ends with a state "End." The following are simple examples of state sequences.

- When the player plays four sixteenth notes within a beat (Figure 3(a)),
  Start -> 1 -> 4 -> 7 -> 10 -> End.
- When the player plays three eighth triplets within a beat (Figure 3(b)),
  Start -> 1 -> 5 -> 9 -> End.
- When the player plays a two-notes chord at the beginning of a beat (Figure 3(c)),
  Start -> 1 -> 1 -> End.
- When the player does not play any note within a beat (Figure 3(d)),
  Start -> End.

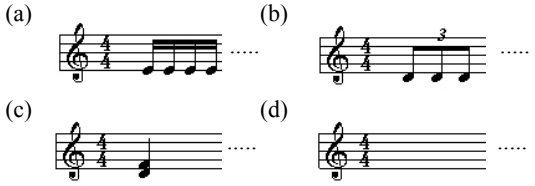

Figure 3. Simple examples of state sequences.

Modeling by a combination of multiple HMMs: Instead of using a single HMM model, we can use multiple HMMs. The advantage of using multiple models is that each model becomes simple. Here we propose a model consisting of four HMMs, each of which represents a different type of rhythm structure within a beat, called *inside-beat type*. We define the following four inside-beat types: sixteenth-note type, eight-triplet type, quarter-note type, and no-note type (Figure 4(b)). The state transition arcs in the model composed of multiple HMMs are simpler than those in a single HMM model (Figure 4(a)).

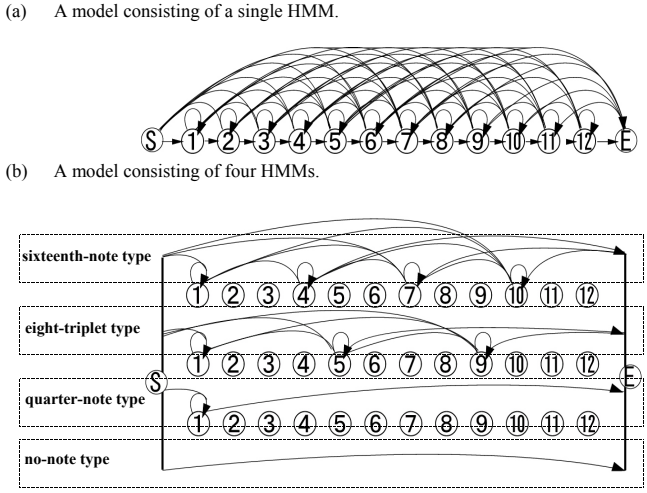

Figure 4. A model consisting of a single HMM and a model consisting of four different HMMs.

Estimation of the optimal sequence of onset times: By concatenating the quarter-note HMMs and using Viterbi algorithm to search for the sequence of hidden-state transitions that maximizes the posterior probability $P(\boldsymbol{\theta}|\boldsymbol{y})$, we can estimate the most probable sequence of onset times throughout a performance. When a performance includes T notes, the observed onset-time sequence can be denoted $\boldsymbol{y}=(y_1, y_2,\ldots, y_T)$. To acquire the optimal state transition sequence, we define $\delta_t(i)$:

$$\delta_t(i) = \max_{\theta_1,\theta_2,\ldots,\theta_{t-1}} P(\theta_1,\theta_2,\ldots,\theta_{t-1};\theta_t = i; y_1, y_2,\ldots, y_t \mid \lambda), \quad (3)$$

where $\delta_t(i)$ is the best score (highest probability) of the state transition sequence $\boldsymbol{\theta}=(\theta_1,\theta_2,\ldots,\theta_t)$ conditioned that the t-th state θ_t is equal to i, and λ denotes a set of all the parameters of the model. The value of the best score satisfies the following recurrent equation:

$$\delta_{t+1}(j) = \max_i \left[\delta_t(i)\, a_{ij}\right] b_j(y_{t+1}). \quad (4)$$

The Viterbi algorithm searches for the optimal sequence of state transitions by scanning paths on a trellis from left to right (Figure. 5).

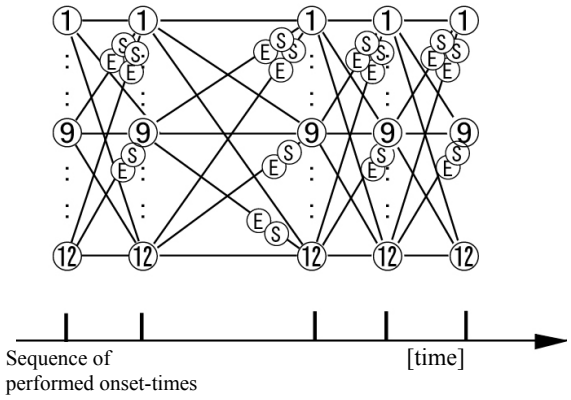

Figure 5. A trellis for finding the optimal state transition.

The horizontal axis in Figure 5 represents the performed onset times and the vertical axis represents the twelve hidden states of the HMM. The optimal state transition sequence can be obtained by choosing one optimal state from 12 possible states while using Eq. (4) to calculate $\delta_t(i)$ from left to right on this trellis. When the state transition passes across a beat line, two dummy states, "End" and "Start," should be inserted.

3. LEARNING MODEL PARAMETERS

By preparing the intended onset times $\boldsymbol{\theta}$ (correct data) and the performed onset times $\boldsymbol{y}$ as training data, it learns a model of onset-time transition $P(\boldsymbol{\theta})$ represented as *aij* and a model of generating the deviation of onset times $P(\boldsymbol{y}|\boldsymbol{\theta})$ represented as $bj(k)$.

Training data

In order to estimate the model parameters *aij* and $bj(k)$, we prepared two sets of training data, artificially generated data and human-performance data. Artificial data is mainly used to confirm that the program works properly.

Artificial data: For each beat we randomly decide whether the inside-beat type is sixteenth-note type or eighth-triplets type. The number of notes in the beat was determined by random numbers between one and six. Then the position of each note was determined randomly: there are four possible positions in the case of sixteenth notes and three possible positions in the case of eighth triplets. Finally, using a normal (Gaussian) distribution with the average 0 and the standard deviation σ, we make the onset times in random data $\boldsymbol{\theta}_a$ deviate, and obtain the artificial data $\boldsymbol{y}_a$. Here the three sets of data were generated by using σ = 10, 20, 30 (1 beat = 480).

Human-performance data: The human-performance data $\boldsymbol{y}_h$ are actual MIDI recordings performed by three human players (guitarists), A, B, and C. Each player played on a MIDI guitar along a fixed-tempo jam session accompaniment. The length of each performance was twelve choruses (1 chorus = 12 bars). Every player performed in two different tempos: one is 120 M.M. and the other is an arbitrary tempo decided by that player. Consequently there were 6 sets of data.

Correct data: For supervised learning, it is necessary to prepare correct data, which is the intended onset times. In the case of artificial data, the onset-time positions $\boldsymbol{\theta}_a$ before adding random deviation can be used as the correct data. For human-performance data, however, we should prepare the correct data by hands. By using the commercial sequencing software (Twelve Tone System, Cakewalk Pro Audio 9) providing a visual piano-roll display, we manually quantized each performed note so its position and duration are proper.

Estimation of model parameters

The HMM parameters *aij* and $bj(k)$ were learned from correct data $\boldsymbol{\theta}$ and the sequence $\boldsymbol{y}$ of performed onset times. Figure 6(a) shows a distribution of $b_5(k)$ learned from artificial data with the standard deviation σ = 20 (1 beat = 480) and Figure 6(b) shows a distribution of $b_5(k)$ learned from the second performance of the player C (we call it performance C2). The $b_5(k)$ learned from artificial data (Figure 6 (a)) becomes a normal distribution because it was generated by normally distributed random numbers. The $b_5(k)$ learned from the performance C2 (Figure 6 (b)) is also nearly normally distributed but is skewed to the right. This shows that a note on the state 5 tended to be delayed from its original position. In fact, in the performance C2 the first note of eighth triplets tended to be played longer and the onset time of the second note consequently tended to be delayed.

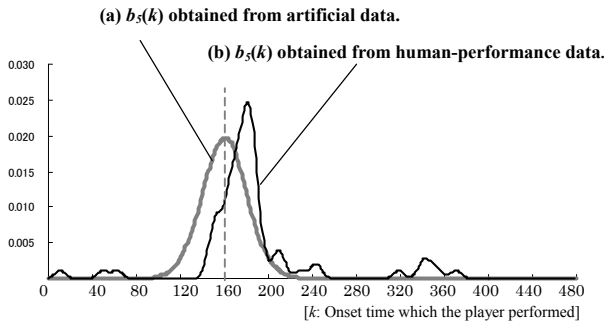

Figure 6: $b_5(k)$ obtained from artificial data and human-performance data.

4. EXPERIMENTAL RESULTS

We tested the proposed method on both artificial data and human-performance data. We evaluated the performance of quantization by using the *correct rate* we defined as follows:

$$(\text{correct rate}) = \frac{(\text{the number of onsets quantized correctly})}{(\text{the number of onsets})} \quad (5)$$

Quantization performance of commercial sequencing software

To evaluate the baseline quantization performance of commercial sequencing software, we specified three different grid intervals (eighth triplet, sixteenth note, and sixteenth triplet) on the software and calculated the correct rate of quantization for each of them (Table 2). The correct rates obtained with other grid intervals were worse than the rates listed in Table 2.

Table 1 also shows the percentages of the four different inside-beat types contained in each performance, which percentages were calculated on the basis of the correct data. The results listed in Tables 1 and 2 show that the correct rates of the sequencing software tended to be lower for the performances containing many sixteenth-note-type beats, such as all the artificial data and the two human-performance data sets A1 and C1. The correct rate for the human-performance data set C2 indicates that the simple quantization with eighth-triplets grids is effective enough for the performance without sixteenth-note-type beats.

Quantization performance of our method with the parameters learned from the same performance data

Table 2 also shows the correct rates for our method using either the single-HMM model or the four-HMM model. For most of the performances, the correct rates for our method were higher than those for the sequencing software. With the artificial data, all the correct rates for both the single-HMM model and the four-HMM model were at least 20 percent higher than those for the sequencing software. These results show that our HMM-based method is effective for both artificial and human-performance data.

We also compared the performances of the single-HMM model and the four-HMM model. With human-performance data, the correct rates for the four-HMM model were higher than those for the single-HMM model; this is because the four-HMM model can represent the observation probability in further detail by learning three different $b_1(k)$ corresponding to the three states (sixteenth-note type, eighth-triplets type, and quarter-note type) at the beginning of a beat. With artificial data, the correct rates for the single-HMM model and the four-HMM model were the same because both the $b_1(k)$ parameter obtained by the single-HMM model and the three different $b_1(k)$ parameters obtained by the four-HMM model were the same normal (Gaussian) distribution, which was used for adding random deviation when generating the artificial data.

Quantization performance of our method with the parameters learned from other performance data.

To evaluate the generalization capability of the method, we calculated the correct rates obtained when the method was used with the following three different sets of the four-HMM model parameters:

(1) parameters learned from the other performance of the same player,

(2) parameters learned from the performances of the other two players, and

(3) parameters learned from the artificial data ($\sigma = 20$).

The results, listed in Table 3, show that with the parameter set (1) the correct rates for players A and B were at most 6.4 percent lower than those obtained when using the parameters learned from the same performance data (at the bottom line of Table 2) and that the correct rates for player C were about 20 percent lower. This is because the performances C1 and C2 had very different styles (C2 did not contain sixteenth-note-type beats), whereas the performances A1 and A2 as well as B1 and B2 were of a similar style.

The results obtained with the parameter sets (2) and (3) show that the correct rates for players A and B were lower than those obtained with the parameter set (1) and that the correct rates for the player C were lower than those obtained with the parameters learned from the same performance data.

These results showed that our method was able to acquire the player's characteristic manner of generating onset-time transition and deviation and that the model parameters learned from a performance of a player can be applied to performances of the same player if the performance styles are similar.

5. CONCLUSION

This paper has described a quantization method using a HMM model of onset-time transition and deviation. This method makes it possible to estimate the intended onset times (without deviation) from the onset times (with deviation) performed along a fixed-tempo jam session accompaniment. Experimental results showed that the proposed model trained using the correct data performed better than commercial sequencing software.

We plan to use this method to create the phrase database for our jam session system automatically and also to extend the method by enabling it to estimate the model parameters from session recordings without correct data.

6. ACKNOWLEDGMENTS

We thank Daisuke Hashimoto, Shinichi Chida, and Yasuhiro Saita, who cooperated with the experiment as players.

7. REFERENCES

[1] M. Hamanaka, M. Goto and N. Otsu, “A Learning Session System: Statistical Modeling of Player’s Reaction,” Information Processing Society of Japan SIG Notes, vol. 2000, no. 19, 2000, pp. 27–34. (in Japanese)

[2] P. Desain and H. Honing, “The Quantization of Music Time: A Connectionist Approach”, Computer Music Journal, vol.13, 1989, pp. 56–66.

[3] H. Katayose and S. Inokuchi, “Intelligent Music Transcription System,” Journal of Japanese Society for Artificial Intelligence, vol.5, no. 1, 1990, pp. 59–66. (in Japanese)

[4] M. Goto and Y. Muraoka, "An Audio-based Real-time Beat Tracking System and Its Applications," Proc. of Intl. Computer Music Conf., 1998, pp. 17–20.

[5] M. Goto, "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds," Journal of New Music Research, 2001 (in press).

[6] R. Dannenberg and B. Mont-Reynaud, “Following an Improvisation in Real Time” Proc. of Intl. Computer Music Conf., 1987, pp. 241–248.

[7] P. Allen and R. Dannenberg, “Tracking Musical Beats in Real Time,” Proc. of Intl. Computer Music Conf., 1990, pp. 140–143.

[8] N. Saitou, M. Nakai, H. Shimodaira and S. Sagayama, “Hidden Markov Model for Restoration of Musical Note Sequence from the Performance,” Information Processing Society of Japan SIG Notes, vol. 99, no. 106, 1999, pp. 27–32. (in Japanese)

Table 1: Percentages of four kinds of beat included in correct data.

	Artificial data			Player A		Player B		Player C	
	σ =10	σ =20	σ =30	A1	A2	B1	B2	C1	C2
Sixteenth-note type	38.8%	38.8%	38.8%	21.4%	5.5%	6.5%	2.7%	37.2%	0.0%
Eighth-triplet type	37.8%	37.8%	37.8%	54.2%	68.7%	58.2%	85.8%	42.1%	39.7%
Quarter-note type	6.9%	6.9%	6.9%	6.8%	10.1%	11.9%	6.4%	5.9%	26.7%
No-note type	16.4%	16.4%	16.4%	17.6%	15.7%	23.3%	5.1%	14.8%	33.8%

Table 2: Performance of commercial sequence software and of our method.

	Artificial data			Player A		Player B		Player C	
	σ =10	σ =20	σ =30	A1	A2	B1	B2	C1	C2
Commercial sequencing software (eighth triplet)	65.6%	58.2%	62.0%	67.6%	85.6%	79.4%	88.6%	57.0%	97.7%
Commercial sequencing software (sixteenth note)	63.9%	67.9%	65.9%	54.5%	37.3%	36.8%	34.7%	70.7%	45.5%
Commercial sequencing software (sixteenth triplet)	77.1%	70.7%	60.8%	57.7%	48.4%	57.8%	51.3%	56.1%	82.8%
The single HMM model of our method	99.6%	95.9%	86.5%	75.9%	84.8%	80.0%	90.5%	85.1%	95.0%
The four HMMs model of our method	99.5%	95.9%	86.5%	82.3%	89.8%	81.4%	92.8%	85.5%	95.7%

Notes: Underlined rates are those for which the proposed method outperforms the commercial sequencing software.

Table 3: Quantization results obtained using the model parameters for other performances.

	Artificial data			Player A		Player B		Player C	
	σ =10	σ =20	σ =30	A1	A2	B1	B2	C1	C2
1) Other performance of the same player	-	-	-	75.9%	85.3%	79.4%	91.6%	55.5%	77.9%
2) Performance of two other players	-	-	-	70.2%	79.0%	59.2%	73.5%	76.9%	93.1%
3) Artificial data	-	-	-	73.3%	53.4%	60.8%	55.2%	82.5%	86.1%