# A Learning-Based Quantization: Unsupervised Estimation of the Model Parameters

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

*Research Fellow of the Japan Society for the Promotion of Science,
** "Information and Human Activity", PRESTO, JST, ***National Institute of Advanced Industrial Science and Technology (AIST), ****University of Tokyo
*email*: m.hamanaka@aist.go.jp

## Abstract

This paper describes a method for organizing onset times performed along a jam-session accompaniment into normalized (quantized) positions in a score so the performance data can be stored in a reusable form. Unlike most previous beat-tracking-related methods that focus 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. Our method solves this problem by using hidden Markov models (HMMs) that model onset-time transition and deviation. The HMM parameters are obtained by unsupervised estimation using the Baum-Welch algorithm and held-out interpolation: they can be derived from only the session recording that we wanted to quantize. Experimental results show that our model performs better than the semi-automatic quantization in commercial sequencing software.

## 1 Introduction

We have been constructing a jam session system (Hamanaka, Goto, and Otsu 2001) 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 phrase database (Fig. 1). Because this database was manually prepared, the system cannot automatically imitate the player's characteristic phrases.

A method of 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 in the score. Because there is no score to follow in improvisation, the normalized positions in a score mean the positions at which a player intended to play. A typical quantization method 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 position. 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.

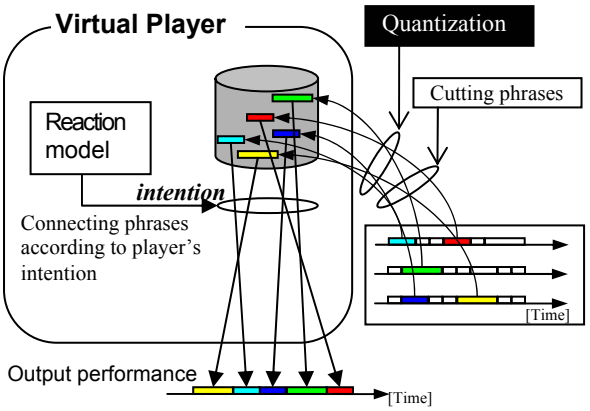



Figure 1: Performance generation using phrase databases.

Several quantization methods have been proposed. Because tempo tracking methods (Dannenberg and Mont-Reynaud 1987), (Allen and Dannenberg 1990), (Vercoe and Puckette 1985) for score following need to use an annotated score as prior knowledge, they cannot quantize improvisations of a jam session. Beat tracking methods (Katayose and Inokuchi 1990), (Goto and Muraoka 1998), (Goto 2001), (Dixon 2001) focus on predicting beat positions and cannot quantize notes which are not in the beat position. A quantization method using connectionist model (Desain and Honing 1989) defines a potential energy that is stable if the ratio of the sum of onset time intervals to the sum of other intervals is an integer. It is not easily applied to various performances, however, because the potential energy is fixed.

On the other hand, quantization methods for automatic transcription (Takeda *et al*. 2003), (Cemgil *et al*. 2000) integrate tempo tracking and quantization. They indicate that a continuous speech recognition framework using a probabilistic model provides a useful approach for estimating tempos and beats and allocating bar lines. However, difficult problems still

remain, such as discrimination between eighth triplets notes and sixteenth notes even if the tempo is known or constant. Quantization in commercial sequencing software is not effective at solving this problem because an onset time that has a large deviation is aligned to an incorrect grid.

The quantization method in our previous work (Hamanaka *et al*. 2001) using HMMs indicates that the performance of the discrimination between eight triplets notes and sixteenth notes improves if we configure the HMM parameters properly. But it could not quantize a human-performance of a new player using appropriate HMM parameters because the HMM parameters were supervised trained with correct data. In this study, we propose an unsupervised estimation method of the model parameters from a session recording using the Baum-Welch algorithm and held-out interpolation. The HMM-based method can achieve proper quantization.

# 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 note onset times because the onset times deviate according to the performer's actions and expression. We can model a process to generate the deviations by using a probabilistic model. The problem of quantization, which acquires the sequence of onset times that the player intended from the sequence of deviating onset times that the player actually performed, can be considered as an inverse problem. This inverse problem can then be solved using the inverse model derived from the model that generates the deviation of onset times.

## 2.1 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 for generating the deviation of onset times can be expressed by a conditional probability $P(\boldsymbol{y} \mid \boldsymbol{\theta})$ (Fig. 2). Using this conditional probability and the prior probability $P(\boldsymbol{\theta})$, the inverse model can be calculated as Eq. (1) according to Bayes' theorem.

$$P(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{P(\boldsymbol{y} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta})}{P(\boldsymbol{y})} \quad (1)$$

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

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P(\boldsymbol{\theta} \mid \boldsymbol{y}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P(\boldsymbol{y} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta}). \quad (2)$$

Because $P(\boldsymbol{y})$ is independent of $\boldsymbol{\theta}$, it can be ignored.

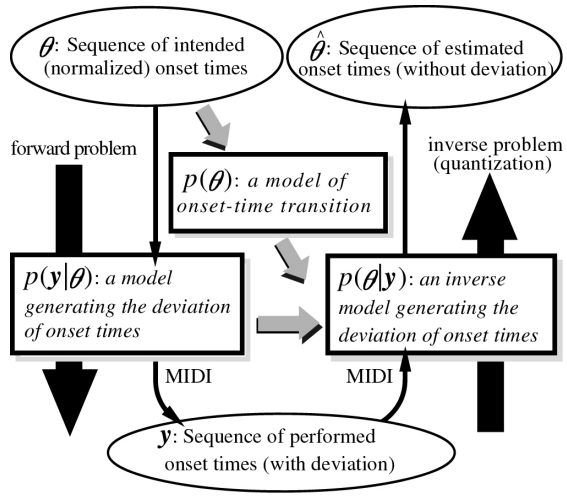



Figure 2: Forward model and inverse model in the quantization problem.

## 2.2 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 the transition sequence of hidden states as a Markov chain. Each hidden state in the state transition sequence then generates an observation value according to the 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 Takeda (Takeda, *et al.* 2003) is limited, our onset-time-based model is suitable for polyphonic performances, such as those that include two-hand piano voicing and guitar arpeggio.

- ***Unit in modeling***

We use a quarter note (beat) as the unit of each HMM, i.e., the temporal length corresponding to each HMM is a quarter note. The reason we use a 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-distant positions within a quarter note duration, while the four notes of the sixteenth notes are located on four equi-distant 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 reduces the calculation time and facilitates the preparation of large data sets 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 for describing performed onset time; it is 1/480 of a quarter note, a value often used in commercial sequencing software. The unit of $i$ is a quantization unit for describing the intended onset time; it is one-twelfth of a quarter note. It can describe both eighth triplets and sixteenth notes.

**Quarter-note hidden Markov model.** Figure 3 shows the HMM used in our study to model a sequence of onset times within a quarter note (beat). All the hidden states of the HMM correspond to the possible positions of the intended onset times, and the observed value that comes from a hidden state corresponds to a performed onset time with deviation.

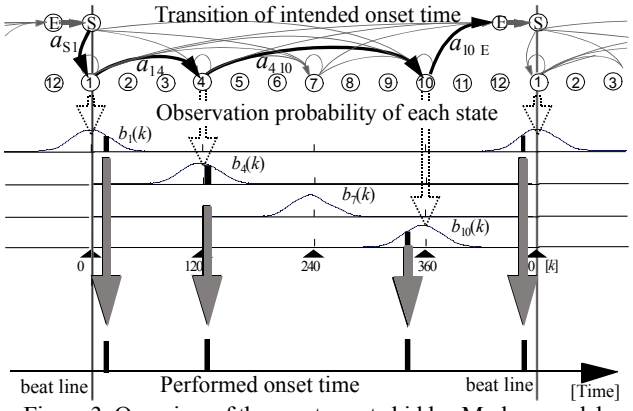
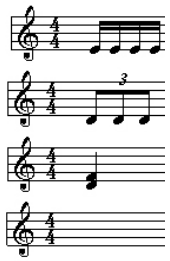


Figure 3: Overview of the quarter-note hidden Markov model.

Onset times in a beat are quantized into 12 positions for hidden states, and into 480 positions for observation values. That is, each HMM component is interpreted as follows.
Hidden state $i$: intended onset time. ($i$ =1, …, 12)
Observation $k$: performed onset time. ($k$ = 1,…, 480)
Transition probability $a_{ij}$: probability that the intended onset time $j$ follows the intended onset time $i$.
Observation probability $b_i(k)$: probability that the performed onset time is $k$ and the intended onset time is $i$.
A state-transition sequence begins with a dummy state "Start" and ends with a state "End". Figure 4 shows simple examples of state sequences.


Start -> 1 -> 4 -> 7 -> 10 -> End.

Start -> 1 -> 5 -> 9 -> End.

Start -> 1 -> 1 -> End.

Start -> End.

Figure 4: Simple example of state sequences.

**Estimation of the optimal sequence of onset times.** By concatenating the quarter-note HMMs and using the Viterbi algorithm that maximizes the posterior probability P( $\boldsymbol{\theta}$ |$\boldsymbol{y}$) to search for the sequence of hidden-state transitions, 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 as $\boldsymbol{y}$=($y_1$, $y_2$,…, $y_T$). To acquire the optimal state transition sequence, we define $\delta_t(i)$ as:

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

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

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

## 2.3 Unsupervised Estimation of Model Parameters

The HMM parameters $a_{ij}$ and $b_i(k)$ were derived from a set of human-performance data using the Baum-Welch algorithm and held-out interpolation. The human-performance data, $\boldsymbol{y}$, comprises actual MIDI recordings performed by three human players (guitarists), A, B, and C. Each player played on a MIDI guitar along with a fixed-tempo jam session accompaniment. Each performance was twelve choruses long (1 chorus = 12 bars). Each player performed at two different tempos: 120 M.M. and an arbitrary tempo decided by the player. Consequently, there were 6 sets of data (A1, A2, B1, B2, C1, and C2).

**Baum-Welch algorithm.** The HMM parameters $a_{ij}$ and $b_i(k)$ were learned from the first half of the human-performance data using the Baum-Welch algorithm.

- ***Set the initial value*** $a^0_{ij}$ ***and*** $b^0_i(k)$ ***properly***.

Let $a^0_{ij}$ be the average $a_{ij}$ derived from the 6 sets of data. Let $b^0_i(k)$ have a normal distribution with mean = (i-1)*40 and standard deviation = 20 (1 beat = 480). The similarity of the distributions of $b_i(k)$ and a normal distribution with mean = ($i$-1)*40 and standard deviation = 20 has been reported previously (Hamanaka *et al*. 2001).

- ***Forward-Backward Algorithm***.

Calculate the forward probability $\alpha_t(i)$ and backward probability $\beta_t(i)$. $\alpha_t(i)$ is the probability of the partial observation sequence from the start of a human-performance to note $t$ given state $i$ at $t$ and the model $\lambda$. $\beta_t(i)$ is the probability of the partial observation sequence from $t$+1 to the end of a human-performance, given state $i$ at t and the model $\lambda$.

- ***Reestimation of model parameter*** $\lambda$

Reestimate $a_{ij}$ and $b_i(k)$ using a recurrence formula as follows. $T$ is the number of notes in the human-performance data.

$$a^{k+1}{}_{ij} = \frac{(\textit{expected number of state-transition from } i \textit{ to } j)}{(\textit{expected number of state-transition from } i)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a^k{}_{ij} b^k{}_j(y_{t+1} \bmod 480) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)} \quad (5)$$

$$b^{k+1}{}_j(k) = \frac{\left(\begin{array}{l}\textit{expected number instants in state } i\\ \textit{and having observation } k\end{array}\right)}{(\textit{expected number instants in state } i)} = \frac{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j) \Big/ \sum_{j=1}^{12} \alpha_t(j)\beta_t(j)}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j) \Big/ \sum_{j=1}^{12} \alpha_t(j)\beta_t(j)} \quad (6)$$

$s.t.(y_{t+1} \bmod 480) = k$

- ***Iteration***

After the reestimation of the model parameters, we have another model, $\hat{\lambda}$, which is more likely than model $\lambda$ to produce the observed onset-time sequence $\boldsymbol{y}$. This means that

$$\mathrm{P}(\boldsymbol{y}|\hat{\lambda}) > \mathrm{P}(\boldsymbol{y}|\lambda). \quad (7)$$

The reestimation process can be continued until no further improvement in P($\boldsymbol{y}|\lambda$) is achieved; that is, until a local maximum is reached.

**Held-Out interpolation.** When using the Baum-Welch algorithm, the HMM parameters generally converge, but the model parameters $b_i(k)$ cannot be reestimated properly when the number of notes in the human-performance data is insufficient. To reestimate

$b_i(k)$ properly, we use the held-out interpolation technique. In the following explanation, we use normalized time $l$ to align the time that corresponds to state $j$ (Eq. (8)).

$$l = \begin{cases} k - 40 \cdot (j-1) & (k - 40 \cdot (j-1) \geq 0) \\ k - 40 \cdot (j-1) + 480 & (k - 40 \cdot (j-1) < 0) \end{cases} \quad (8)$$

We perform linear interpolation of $b_j(l)$ ( $j$ = 1, 2, …, 12) by an interpolation factor $\zeta$ ($0 \leq \zeta \leq 1$).

$$\hat{b}_j(l) = \zeta\, \bar{b}(l) + (1 - \zeta)\, b_j(l) \quad \text{(k=0, 1, …, 479)} \quad (9)$$

$\bar{b}(l)$ is an average of $b_i(l)$.

$$\bar{b}(l) = \frac{1}{12} \sum_{j=1}^{12} b_j(l) \quad (10)$$

$\hat{b}_j(l)$ is a more accurate estimate than $b_i(l)$ if we properly configure the interpolation factor, $\zeta$ (Efron and Morris 1977).

We estimate $\zeta$ by using the EM algorithm from the second half of the human-performance data. Equation (11) is for reestimation of $\zeta$.

$$\hat{\zeta} = \frac{1}{T} \sum_{t=1}^{T} \frac{\zeta \times b_j(l)}{(1-\zeta) \times \bar{b}_j(l) + \zeta \times b_j(l)} \quad (11)$$

The HMM parameters $a_{ij}$ and $b_i(k)$ were derived by iterating the Baum-Welch algorithm and held-out interpolation.

# 3. Experimental Results

We evaluated the performance of quantization using the rate of correct quantization, which we defined as follows:

$$(\text{correct rate}) = \frac{(\text{the number of onsets quantized correctly})}{(\text{the number of onsets})} \quad (12)$$

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 rate of correct quantization for each (Table 1). The correct rates obtained with other grid intervals were worse than the rates listed in Table 1. Table 1 also shows the correct rates for our method. For all of the performances the correct rates for our method were higher than 70 percent. Our method performed better than the commercial sequencing software on the human-performance data (A2 and C1) which contain eight triplet bars in half of it and sixteenth bars in the other half. The simple quantization with eight triplet grids was effective on most of the human-performance data (A1, B2, and C2) which contain eight triplet bars in most of it. The results showed that our method is effective in typical cases where the rhythm structure changes frequently during human-performance data (i.e., the ratio of eight triplet bars and sixteenth bars is not known in advance).

# 4. Conclusion

This paper has described a quantization method that uses HMMs for modeling 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 together with a fixed-tempo jam session accompaniment. Experimental results showed that the proposed method that trains the HMMs with the human-performance data using the Baum-Welch algorithm and held-out interpolation is effective when the rhythm structure changes frequently.

We plan to use this method to automatically generate the phrase database for our jam session system.

# References


Hamanaka, M., Goto, M., and Otsu, N. 2001. "Learning-Based Jam Session System for a Guitar Trio." *Proceedings of the International Computer Music Conference.* pp. 467-470.

Dannenberg, R. B. and Mont-Reynaud, B. 1987. "Following an Improvisation in Real Time." *Proceedings of the International Computer Music Conference.* pp. 241-248.

Allen, P. and Dannenberg, R. B. 1990. "Tracking Musical Beats in Real Time." *Proceedings of the International Computer Music Conference.* pp. 140-143.

Vercoe, B.L. and Puckette, M. 1985. "The synthetic rehearsal: Training the synthetic performer." *Proceedings of the International Computer Music Conference.* pp. 275-278.

Katayose, H. and Inokuchi, S. 1990. "Intelligent Music Transcription System." *Journal of Japanese Society for Artificial Intelligence* 5(1): 59-66. (in Japanese)

Goto, M. and Muraoka, Y. 1998. "An Audio-based Real-time Beat Tracking System and Its Applications." *Proceedings of the International Computer Music Conference.* pp. 17-20.

Goto, M. 2001. "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds." *Journal of New Music Research* 30(2):159-171.

Dixon, S. 2001. "An Interactive beat tracking and visualization system." *Proceedings of the International Computer Music Conference.* pp. 215-218.

Desain, P. and Honing, H. 1989. "The Quantization of Music Time: A Connectionist Approach." *Computer Music Journal* 13(3):56-66.

Takeda, H., Nishimoto, T., Shinoda, K., and Sagayama, S. 2003. "Score Estimation from Polyphonic MIDI Performance with Probabilistic Models." *Information Processing Society of Japan SIG Notes* 2003(48): 21-26. (in Japanese)

Cemgil, A. T., Desain, P., and Kappen, H. J. 2000. "Rhythm quantization for transcription." *Computer Music Journal* 24(2):60-76.

Hamanaka, M., Goto, M., Asoh, H., and Otsu, N. 2001. "A Learning-Based Quantization: Estimation of Onset Times in a Musical Score." *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics 2001* 10(1):pp. 374-379.

Efron, E. and Morris, C. 1977. "Stein's Paradox in Statistics." *Scientific American* 20(1): pp. 451-468.


Table 1: Performance of commercial sequencing software compared with our method.

| | Player A | | Player B | | Player C | |
|---|---|---|---|---|---|---|
| | A1 | A2 | B1 | B2 | C1 | C2 |
| Commercial sequencing software (eighth triplet) | 85.6% | 67.6% | 79.4% | 88.6% | 57.0% | 97.7% |
| Commercial sequencing software (sixteenth note) | 37.3% | 54.5% | 36.8% | 34.7% | 70.7% | 45.5% |
| Commercial sequencing software (sixteenth triplet) | 48.4% | 57.7% | 57.8% | 51.3% | 56.1% | 82.5% |
| Our HMM-based quantization method | 74.7% | 78.4% | 72.1% | 78.2% | 84.7% | 89.4% |