Ali Taylan Cemgil,* Peter Desain,† and Bert Kappen*
*Stitching Neurale Netwerken (SNN)
Department of Medical Physics and Biophysics
University of Nijmegen
Geert Grooteplein 21 cpk1-231
NL 6525 EZ Nijmegen, The Netherlands
cemgil@mbfys.kun.nl
http://www.mbfys.kun.nl/~cemgil
bert@mbfys.kun.nl
† Nijmegen Institute for Cognition and
Information (NICI)
University of Nijmegen
P.O. Box 9104
NL 6500 HE Nijmegen, The Netherlands
desain@nici.kun.nl

Rhythm Quantization for Transcription

Automatic music transcription is the extraction of an acceptable musical description from performed music. Interest in this problem is motivated by the desire to design a program that automatically notates a performance. In general, when directly operating on an acoustical recording of polyphonic music (polyphonic pitch tracking), this task has proven to be a difficult and as-yet unsolved problem. Surprisingly, even a simpler subtask still remains difficult, namely, producing an acceptable notation from a list of onset times (e.g., a sequence of MIDI events) under unconstrained performance conditions.

Although quantization of a "mechanical" performance is rather straightforward, the task becomes increasingly difficult in the presence of expressive variations, which can be thought of as systematic deviations from a pure mechanical performance. In such unconstrained performance conditions, two types of systematic deviations from exact values occur. At small time scales, notes can be played accented or delayed. At large scales, tempo can vary; for example, the player can accelerate (or decelerate) during performance, or slow down (ritard) at the end of the piece. In any case, these timing variations usually obey a certain structure, since they are mostly intended by the performer. Moreover, they are linked to several attributes of the performance such as meter, phrase, form, style, etc. (Clarke 1985). To devise a general computational model (i.e., a performance model) that takes all these factors into account is quite difficult.

Another observation important for quantization is that we perceive a rhythmic pattern not as a sequence of isolated onsets, but rather as a perceptual entity comprised of onsets. This also suggests that attributes of neighboring onsets such as duration, timing deviation, etc. are correlated in some way.

This correlation structure is not fully exploited in commercial music software that performs automated music transcription and score typesetting. The usual approach taken is to assume a constant tempo throughout the piece and to quantize each onset to the nearest grid point implied by the tempo and a suitable prespecified minimum note duration (e.g., eighth, sixteenth, etc.). Such a grid-quantization scheme implies that each onset is quantized to the nearest grid point independently of its neighbors; thus, all of its attributes are assumed to be independent, and hence the correlation structure is not employed. The consequence of this restriction is that users are required to play along with a fixed metronome and without any expression. The quality of the resulting quantization is only satisfactory if the music is performed according to the assumptions made by the quantization algorithm. In the case of grid quantization, this is a mechanical performance with small and independent random deviations.

More elaborate models for rhythm quantization in-

Computer Music Journal, 24:2, pp. 60–76, Summer 2000
© 2000 Massachusetts Institute of Technology.

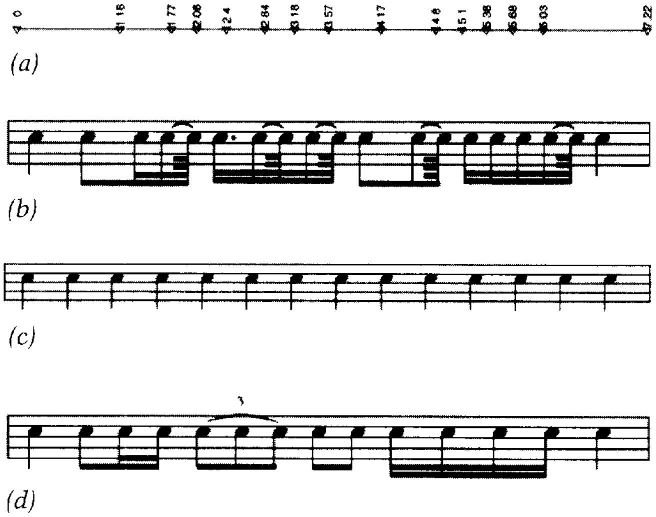
directly take the correlation structure of expressive deviations into account. In one of the first quantization attempts, Longuet-Higgins (1987) used the hierarchical structure of musical rhythms. Desain, Honing, and de Rijk (1992) used a relaxation network in which pairs of time intervals were attracted to simple integer ratios. Pressing and Lawrence (1993) used several template grids, and compared both onsets and inter onset intervals (IOIs) to the grid, selecting the best quantization according to some distance criterion. The Kant system by Agon and colleagues (1994) developed at IRCAM used more sophisticated heuristics, but was in principle similar to the method of Pressing and Lawrence (1993).

The main criticism of these models is that the assumptions about the expressive deviations are implicit and are usually hidden in the model, and thus it is not always clear how a particular design choice affects the overall performance for a range of musical styles. Moreover, it is not directly possible to use experimental data to tune model parameters to enhance the quantization performance. In this article, we describe a method for quantization of onset sequences. We begin by stating the transcription problem and defining the relevant terminology. Using the Bayesian framework, we describe probabilistic models for expressive deviation and notation complexity, and show how different quantizers can be derived from them. Finally, we train the resulting model on experimental data obtained from a psychoacoustical experiment, and compare its performance to simple quantization strategies.

Problem Description

We defined automated music transcription as the extraction of an acceptable description (music notation) from a music performance. In this study, we concentrate on a simplified problem, where we assume that a list of onset times is provided, excluding tempo, pitch, or note-duration information. Given any sequence of onset times, we can in principle easily find a notation (i.e., a sequence of rational numbers) to describe the timing information arbitrarily well. Equivalently, we can find several scores describing the same rhythmic figure for any given error rate, where by "error" we mean some distance between onset times of the performed rhythm and the mechanical performance (e.g., as would be played by a computer).

Figure 1. Different quantizations of an onset sequence. A performed onset sequence example (a). A too-accurate quantization (b); although the resulting notation represents the performance well, it is unacceptably complicated. Too-simple notation (c); although the notation is simpler, it offers a very poor description of the rhythm. The desired quantization (d) balances accuracy and simplicity.



Consider the performed simple rhythm in Figure 1a (from Desain and Honing 1991). A very fine-grid quantizer produces a result similar to Figure 1b. Although this is an accurate representation, the resulting notation is far too complex. Another extreme case is the notation in Figure 1c. Although this notation is simple, it is unlikely that it is the intended score, since this would imply unrealistic tempo changes during the performance. Musicians would probably agree that the "smoother" score shown in Figure 1d is a better representation.

This example suggests that a *good score* must be "easy" to read while representing the timing information accurately. This is apparently a trade-off, and a quantization schema must balance these two conflicting requirements. In the following section, we define more concretely what we mean by a simple score and an accurate representation.

Rhythm-Quantization Problem

Definitions

In this section, we give formal definitions of the terms that we use in the derivations to follow. A performed rhythm is denoted by a sequence $[t_i]$, where each entry is the time of occurrence of an onset. We denote a set with the typical element x_i as $\{x_i\}$ If the elements are ordered (e.g., to form a string), we use $[x_i]$. For example, the performed rhythm in Figure 1a is represented by $t_1 = 0$, $t_2 = 1.18$, $t_3 = 1.77$, $t_4 = 2.06$, etc. We also use the terms "performance" and "rhythm" interchangeably when we refer to an onset sequence.

An important subtask in transcription is *tempo tracking*, i.e., the induction of a sequence of points (beats) in time, which coincides with the human sense of rhythm (e.g., foot tapping) while listening to music. Significant research has already been done on psychological and computational modeling aspects of this behavior (Large 1995; Toiviainen 1999).

We call such a sequence of beats a tempo track, and denote it by $\vec{\tau} = [\tau_j]$, where τ_j is the time at which the *j*th beat occurs. We note that for automatic transcription, $\vec{\tau}$ is to be estimated from $[t_i]$.

Once a tempo track $\vec{\tau}$ is given, the rhythm can be divided into a sequence of segments, each of duration $t_j - t_{j-1}$ The *j*th segment contains K_j onsets, which we enumerate by $k = 1 \ldots K_j$. The onsets in each segment are normalized and denoted by $\mathbf{t}_j = [t_j^k]$ for all $\tau_{j-1} \le t_i < \tau_j$ where

$$t_j^k = \frac{t_i - \tau_{j-1}}{\tau_j - \tau_{j-1}}. \qquad (1)$$

Note that this is merely a re-indexing from a single index *i* to a double index (k,j). When an argument applies to all segments, we will drop the index *j*. In other words, the onsets are scaled and translated such that an onset just at the end of the segment is mapped to unity and another just at the beginning to zero. The segmentation of a performance is given in Figure 2.

Once a segmentation is given, the quantization process reduces to mapping onsets to locations, which can be described by simple rational numbers. Because Western musical notation is generated by recursive subdivisions of a whole note, it is also convenient to generate possible onset-quantization locations by regular subdivisions. We let $S = [s_i]$ denote a subdivision schema, where $[s_i]$ is a sequence of small prime numbers. Possible quantization locations are generated by subdividing the unit interval [0,1]. At each new iteration *i*, the intervals already generated are divided further into s_i equal parts, and the resulting endpoints are added to a set C . Note that this procedure places the quantization locations on a grid of points c_n, where two neighboring grid points have the distance $1/\prod_i s_i$. We denote the first iteration number at which the grid point c is added to C as the *depth* of c with respect to S. This number is denoted as $d(c|S)$.

As an example, consider the subdivision $S = [3,2,2]$. The unit interval is divided first into three equal pieces, then the resulting intervals into two, and so on. At each iteration, generated endpoints are added to the list. In the first iteration, 0, 1/3, 2/3, and 1 are added to the list. In the second iteration, 1/6, 3/6, and 5/6 are added, etc. The resulting grid points (filled circles) are depicted in Figure 3. The vertical axis corresponds to $d(c|S)$.

If a segment $\mathbf{t}$ is quantized (with respect to S, the result is a K-dimensional vector with all entries on some grid points. Such a vector we call a *code vector*, and denote as $\mathbf{c} = [c_k]$. i.e., $\mathbf{c} \in C \times C \ldots \times C = C^K$. We call a set of code vectors a *code book*. Since all entries of a code vector coincide with some grid points, we can define the depth of a code vector as

$$d(\mathbf{c} \,\Big|\, S) = \sum_{c_k \in \mathbf{c}} d(c_k \mid S). \qquad (2)$$

A score can be viewed as a *concatenation* of code vectors $\mathbf{c}_j$. For example, the notation in Figure 4a can be represented by a code-vector sequence as in Figure 5. Note that the representation is not unique: both code-vector sequences represent the same notation.

Performance Model

As described in the introduction, musical performances are subject to several types of systematic deviations. In the absence of such deviations, every score would have only one possible interpretation. Clearly, two natural performances of a piece of music are never the same. Even performances of very short rhythms show deviations from a strict me-

*Figure 2. Segmentation of a performance by a tempo track (*vertical *dashed lines): $\vec{\tau}$ = [0.0, 1.2, 2.4, 3.6, 4.8, 6.0, 7.2, 8.4]. The resulting segments are* **t** *= [0],* **t** *= [0.475,0.717], etc.*

Figure 3. Depth of gridpoint c by subdivision schema S = [3,2,2].

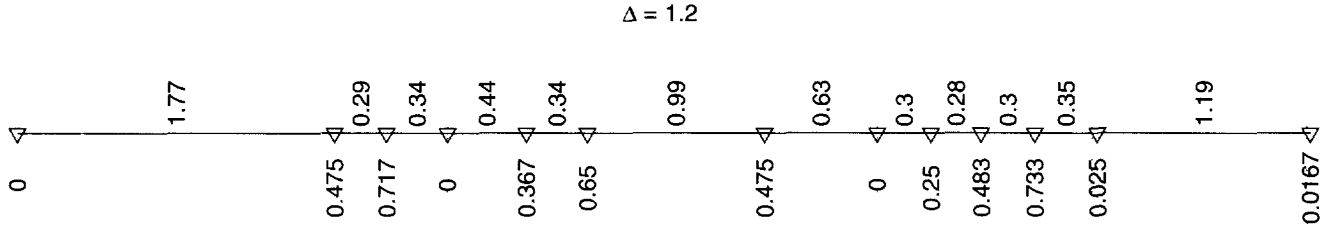
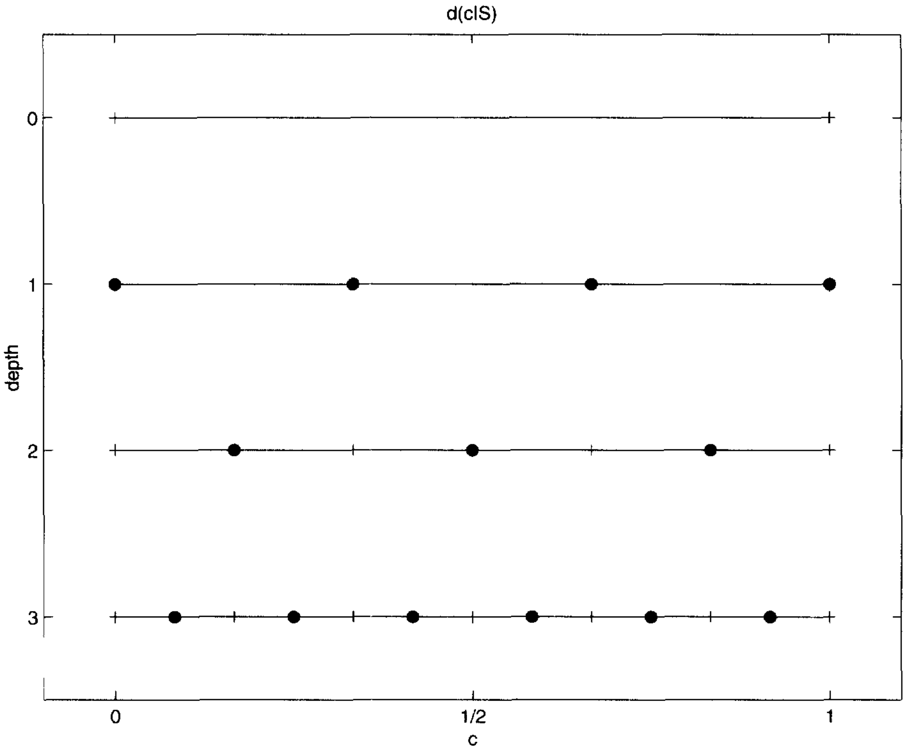


Figure 2

Figure 3



chanical performance. In general terms, a *performance model* is a mathematical description of such deviations; it describes how likely a score is mapped to a performance (see Figure 4). Before we describe a probabilistic performance model, we briefly review a basic theorem of probability theory.

Bayes's Theorem

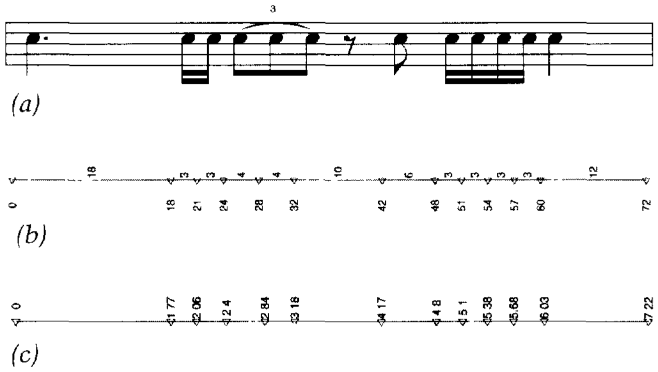
The joint probability $p(A,B)$ of two random variables A and B defined over the respective state spaces S_A and S_B can be factorized in two ways:

$$p(A,B) = p(B \mid A)p(A) = p(A \mid B)p(B) \qquad (3)$$

where $p(A \mid B)$denotes the conditional probability of A given B for each value of B, this is a probability distribution over A. Therefore $\sum_A p(A \mid B) = 1$ for any fixed B. The marginal distribution of a variable can be found from the joint distribution by summing over all states of the other variable:

$$p(A) = \sum_{B \in S_B} p(A,B) = \sum_{B \in S_B} p(A \mid B)p(B). \qquad (4)$$

*Figure 4. A simplified schema of onset quantization: a notation (a) defines a score (b) which places onsets on simple rational points with respect to a tempo track (**vertical** dashed lines). The performer "maps" (b) to a performance (c). This process is not deterministic; in every new performance of this score, a (slightly) different performance would result. A performance model is a description of this stochastic process. The task of the transcriber is to recover both the tempo track and the onset locations in (b), given (c).*



It is understood that summation is to be replaced by integration if the state space is continuous. Bayes's theorem results from equations 3 and 4:

$$p(B \mid A)=\frac{p(A \mid B)p(B)}{\sum_{B\in S_B} p(A \mid B)p(B)} \propto p(A \mid B)p(B). \qquad (5)$$

The proportionality follows from the fact that the denominator does not depend on B, because B is already summed over. This rather simple-looking equation has surprisingly far-reaching consequences and can be directly applied to quantization. Consider the case that B is a score and S_B is the set of all possible scores. Let A be the observed performance. Then equation 5 can be written as

$$\begin{gathered} p(\textit{score} \mid \textit{performance}) \propto p(\textit{performance} \mid \textit{score}) \times p(\textit{score}) \\ \textit{posterior} \propto \textit{likelihood} \times \textit{prior}. \end{gathered} \qquad (6)$$

The intuitive meaning of this equation can be better understood if we think of quantization as a score-selection problem. Since there is usually not a single correct notation for a given performance, several possibilities will exist. The most reasonable choice is to select the score **c** which has the highest probability given the performance **t**. Technically, we name this probability distribution as the posterior $p(\mathbf{c}|\mathbf{t})$. The name posterior comes from the fact that this quantity appears after we observe the performance **t**. Note that the posterior is a function over **c**, and assigns a number to each notation after we fix **t**. We look for the notation **c** that maximizes this function. Bayes's theorem tells us that the posterior is proportional to the product of two quantities, the likelihood $p(\mathbf{t}|\mathbf{c})$ and the prior $p(\mathbf{c})$. Before we explain the interpretation of the likelihood and the prior in this context, we first summarize the ideas in compact notation as

$$p(\mathbf{c} \mid \mathbf{t}) \propto p(\mathbf{t} \mid \mathbf{c})p(\mathbf{c}). \qquad (7)$$

The best code vector $\mathbf{c}^*$ is given by

$$c^* = \arg\max_{c\in C^K} p(c \mid t). \qquad (8)$$

In technical terms, this problem is called a *maximum aposteriori* (MAP) estimation problem, and $\mathbf{c}^*$ is called the MAP solution of this problem. We can also define a related quantity L (minus log posterior) and try to minimize this quantity rather then maximizing equation 7 directly. This simplifies the form of the objective function without changing the locations of local extrema, because $\log(x)$ is a monotonically increasing function. The quantity L is defined as

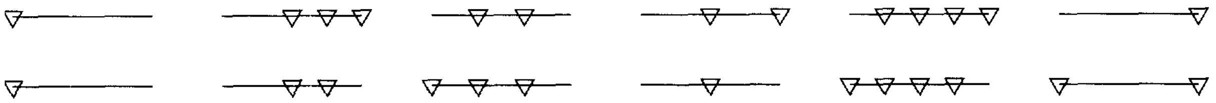
$$L = -\log p(\mathbf{c} \mid \mathbf{t}) \propto -\log p(\mathbf{t} \mid \mathbf{c}) + \log \frac{1}{p(\mathbf{c})}. \qquad (9)$$

The $-\log p(\mathbf{t} \mid \mathbf{c})$ term in equation 9, which is the minus logarithm of the likelihood, can be interpreted as a distance measuring how far the rhythm **t** is played from the perfect mechanical performance **c**. For example, if $p\{t \mid c)$ is of the form $\exp(-(t - c)^2)$, then $-\log p(t \mid c)$ would be $(t - c)^2$, the square of the distance from **t** to **c**. This quantity can be made arbitrarily small if we use a fine grid, however, as mentioned in the introduction. This would eventually result in a complex notation.

However, a suitable prior distribution prevents this undesired result. The $\log \frac{1}{p(\mathbf{c})}$ term, which is large when the prior probability $p(\mathbf{c})$ of the code vector is small, can be interpreted as a complexity term that penalizes complex notations. The best quantization balances these two terms in an optimal way. The precise form of the prior will be discussed later.

The form of a performance model, i.e., the likelihood, can be in general very complicated. However, in this article we consider a subclass of performance models where the expressive timing is assumed to be an additive-noise component that depends on c. The model is given by

Figure 5. Two equivalent representations of the notation in Figure 3a by a code-vector sequence. Here, each horizontal line segment represents one vector of length 1 beat. The endpoint of one vector is the same point in time as the beginning of the next vector. Note that the only difference between two equivalent representations is that some beginning points and endpoints are swapped.



$$\mathbf{t}_j = \mathbf{c}_j + \varepsilon_j \qquad (10)$$

where ε_j is a vector that denotes the *expressive timing deviation*. In this article we assume that ε_j is normally distributed with zero mean and covariance matrix $\Sigma_\varepsilon(\mathbf{c})$, i.e., the correlation structure depends upon the code vector. We denote this distribution as $\varepsilon \propto N(0, \Sigma_\varepsilon(\mathbf{c}))$. Note that when ε is the zero vector, $\Sigma_\varepsilon \to \mathbf{0}$, the model reduces to a so-called mechanical performance.

Example 1: Scalar Quantizer (Grid Quantizer)

We now provide a simple example that applies these ideas to quantization. Consider a one-onset segment $\mathbf{t} = [0.45]$. Suppose we wish to quantize the onset to one of the endpoints, i.e., we are using effectively the code book $\mathbf{C} = \{[0],[1]\}$. The obvious strategy is to quantize the onset to the nearest grid point (e.g., a grid quantizer), and so the code vector $\mathbf{c} = [0]$ is chosen as the winner.

The Bayesian interpretation of this decision can be demonstrated by computing the corresponding likelihood $p(\mathbf{t} \mid \mathbf{c})$ and the prior $p(\mathbf{c})$. It is reasonable to assume that the probability of observing a performance t given a particular c decreases with the distance $|\mathbf{t} - \mathbf{c}|$. One such probability distribution having this property is the normal (Gaussian) distribution. Because there is only one onset, the dimension $K = 1$, and the likelihood is given by

$$p(t \mid c) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-c)^2}{2\sigma^2} \right). \qquad (11)$$

If both code vectors are equally probable, a "flat" prior can be chosen, i.e., $p(\mathbf{c}) = [1/2,1/2]$. The resulting posterior $p(\mathbf{c} \mid \mathbf{t})$ is plotted in Figure 6. The decision boundary occurs at $t = 0.5$, where $p(\mathbf{c}_1 \mid \mathbf{t}) = p(\mathbf{c}_2 \mid \mathbf{t})$. The winner is given as in equation 8:

$$c^* = \arg\max_c p(c \mid t). \qquad (12)$$

Different quantization strategies can be implemented by changing the prior. For example, if $\mathbf{c} = [0]$ is assumed to be less probable, we can choose another prior, e.g., $p(\mathbf{c}) = [0,3,0.7]$. In this case, the decision boundary shifts from 0.5 toward 0.0, as expected.

Example 2: Vector Quantizer

Assigning different prior probabilities to notations is only one way of implementing different quantization strategies. Other decision regions can be implemented by varying the conditional probability distribution $p(\mathbf{t}|\mathbf{c})$. In this section we demonstrate the flexibility of this approach for quantization of groups of onsets.

Consider the segment $\mathbf{t} = [0.45,0.52]$ depicted in Figure 7. Suppose we wish to quantize the onsets again only to one of the endpoints, i.e., we are using effectively the code book $\mathbf{C} = \{[0,0],[0,1],[1,1]\}$. The simplest strategy is to quantize every onset to the nearest grid point (e.g., a grid quantizer) and so the code vector $\mathbf{c} = [0,1]$ is the winner. However, this result might be not be desirable, since the inter-onset interval (IOI) has increased more than 14 times (from 0.07–1.0). It is less likely that a human transcriber would make this choice, since it is perceptually not very realistic.

We could try to solve this problem by employing another strategy. If $\delta = t_2 - t_1 > 0.5$, we use the code vector [0,1]. If $\delta \leq 0.5$, we quantize to one of the code vectors [0,0] or [1,1], depending on the average of the onsets. Using this strategy, the quantization of [0.45,0.52] yields [0,0].

Although considered to be different in the literature, both strategies are just special cases that can be derived from Equation 9 by making specific choices about the correlation structure (covariance

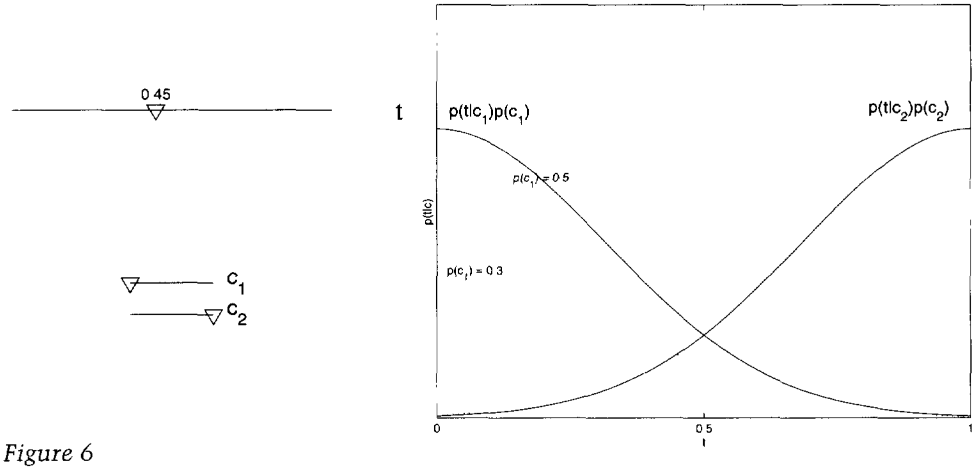


Figure 6

Figure 6. Quantization of an onset as Bayesian inference. When $p(\mathbf{c}) = [1/2,1/2]$, at each t, the posterior $p(\mathbf{c}|\mathbf{t})$ is proportional to the solid lines, and the decision boundary is at $t = 0.5$. When the prior is changed to $p(\mathbf{c}) = [0.3,0.7]$ (dotted lines), the decision boundary moves toward 0.

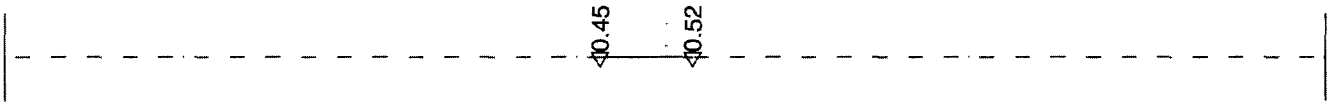


Figure 7

Figure 7. Two onsets.

matrix Σ_ε) of expressive deviations. The first strategy assumes that the expressive deviations of both onsets are independent of each other. This is apparently not a realistic model for timing deviations in music. The latter corresponds to the case where onsets are linearly dependent; it was assumed that $t_2 = t_1 + \delta$, and only δ and t_1 were considered in quantization. This latter operation is merely a linear transformation of onset times, and is implied by the implicit assumption about the correlation structure. Indeed, some quantization models in the literature focus directly on IOIs rather then on onset times.

More general strategies, which can be difficult to state verbally, can be specified by different choices of Σ_ε and $p(\mathbf{c})$. Some examples for the choice $\sum_\varepsilon = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ and constant $p(\mathbf{c})$ are depicted in Figure 8. The ellipses denote the set of points that are equidistant from the center, and the covariance matrix $\sum_\varepsilon$ determines their orientation. The lines denote the decision boundaries. The interested reader is referred to the work of Duda and Hart (1973) for a discussion of the underlying theory.

Likelihood for the Vector Quantizer

For modeling the expressive timing e in a segment containing K onsets, we propose the following parametric form for the covariance matrix:

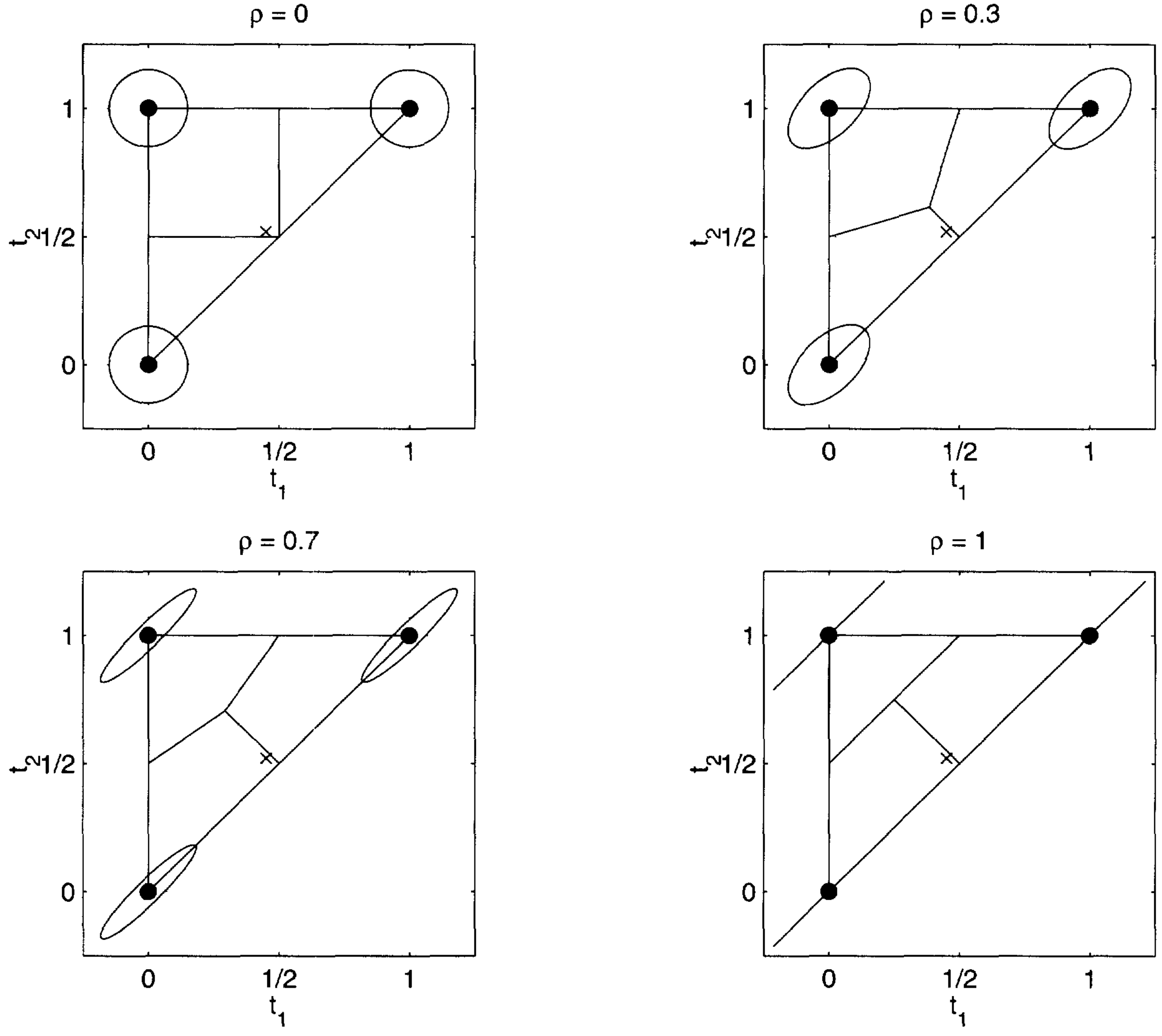
$$\sum\nolimits_\varepsilon(\mathbf{c}) = \sigma^2 \begin{pmatrix} 1 & \rho_{1,2} & \cdots & \rho_{1,K} \\ \rho_{1,2} & 1 & \rho_{n,m} & \vdots \\ \vdots & \rho_{n,m} & \ddots & \vdots \\ \rho_{1,K} & \cdots & \cdots & 1 \end{pmatrix} \quad (13)$$

where

$$\rho_{n,m} = \eta \exp(-\frac{\lambda^2}{2}(c_m - c_n)^2). \quad (14)$$

Here, c_m and c_n are two distinct entries (grid points) of the code vector $\mathbf{c}$, and η is a parameter between −1 and 1 that adjusts the correlation strength between two onsets. The other parameter, λ, adjusts the correlation as a function of the distance between entries in the code vector. When λ is zero, all entries are correlated by the same

Figure 8. Tiling for choices of $p$and constant $p(\mathbf{c})$. Onset quantization (i.e., grid quantization) used by many commercial notation packages corresponds to the case where $\rho = 0$. The IOI quantization appears when $\rho \rightarrow 1$. Note that different correlation structures imply different quantization decisions, not necessarily onset or IOI quantization. The cross corresponds to the rhythm $\mathbf{t} = [0.45,0.52]$.



amount, namely η. When λ is large, the correlation rapidly approaches zero with increasing distance.

This particular choice for $p(\varepsilon)$reflects the observation that onsets that are close to each other tend to be highly correlated. This can be interpreted as follows: if the onsets are close to each other, it is easier to quantify the IOI and then select an appropriate translation for the onsets by keeping the IOI constant. If the grid points are far away from each other, the correlation tends to be weak (or sometimes negative), which suggests that onsets are quantized independently of each other. In the following section, we empirically verify this claim.

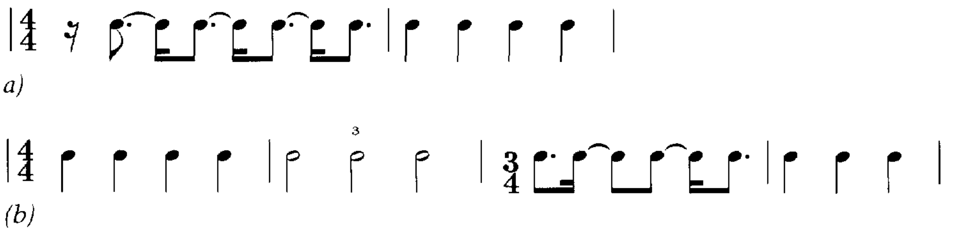
Prior for the Vector Quantizer

The choice of the prior $p(\mathbf{c})$ reflects the complexity of code vector $\mathbf{c}$. In this article, we propose a complexity measure from a probabilistic point of view. The complexity of a code vector $\mathbf{c} = [c_i]$ is determined in this measure by the depth of c_i with respect to the beat (see equation 2) and the time signature of the piece (see Figure 9).

The prior probability of a code vector with respect to S is chosen as

$$p(\mathbf{c} \mid S) \propto exp(-\gamma \mathrm{d}(\mathbf{c} \mid S)). \qquad (15)$$

Figure 9. Complexity of a notation: when no other context is available, both onset sequences will sound the same (a); however, the first notation is more complex (b). The assumed time signature determines the complexity of a notation.



Note that if $\gamma = 0$, the depth of the code vector has no influence upon its complexity. If it is large $((\gamma \approx 1))$, only very simple rhythms get reasonable probability mass. This choice is also in accordance with intuition and experimental evidence: simpler rhythms are more frequently used than complex ones. The marginal prior of a code vector is found by adding all possible subdivision schemes:

$$p(\mathbf{c}) = \sum_{S} p(\mathbf{c} \mid S) p(S) \tag{16}$$

where $p(S)$ is the prior distribution of subdivision schemata.

For example, one can select possible subdivision schemas as $S_1 = [2,2,2]$, $S_2 = [3,2,2]$, and$S_3 = [2,3,2]$. If we have a preference toward the time signature (4/4), the prior can be taken as $p(S) = [1/2,1/4,1/4]$. In general, this choice should reflect the relative frequency of time signatures. We propose the following form for the prior of $S = [s_i]$:

$$p(S) \propto \exp(-\xi \sum_{i} w(s_i)) \tag{17}$$

where $w(s_i)$ is a simple weighting function given in Table 1. This form favors subdivisions by small prime numbers, reflecting the intuition that rhythmic subdivisions by prime numbers such as 7 or 11 are far less common than subdivisions such as 2 or 3. The parameter ξ distributes probability mass over the primes. When $\xi = 0$, all subdivision schemata are equally probable. As ξ increases without limit in the positive direction, only subdivisions with $s_i = 2$ have nonzero probability.

Table 1. $w(s_i)$

s_i	2	3	5	7	11	13	17	o/w
$w(s_i)$	0	1	2	3	4	5	6	∞

Verification of the Model

To choose the likelihood $p(\mathbf{t} \mid \mathbf{c})$ and the prior $p(\mathbf{c})$ in a way that is perceptually meaningful, we analyzed data obtained from a psychoacoustical experiment in which ten well-trained subjects (nine conservatory students and a conservatory professor) participated (Desain et al. 1999). The experiment consisted of a perception task and a production task.

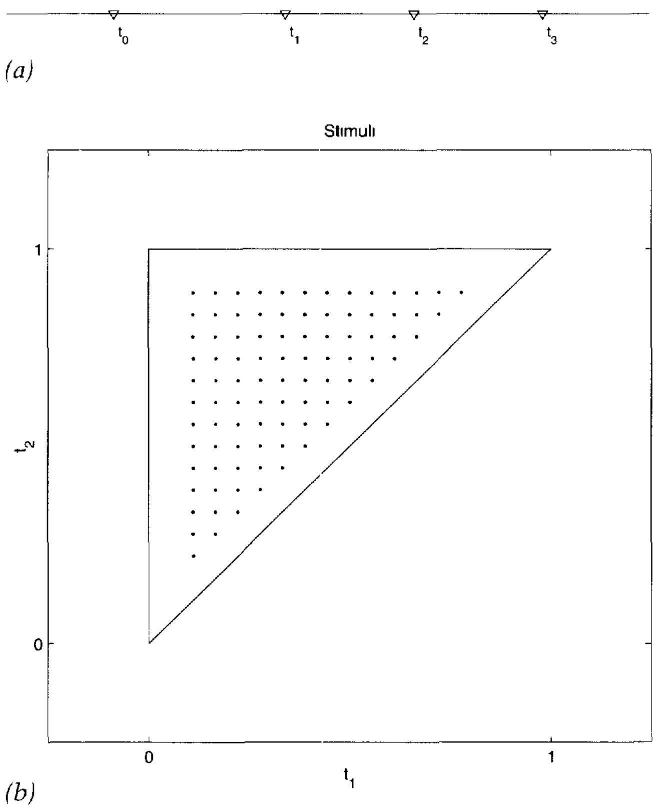
Perception Task

In the perception task, the subjects were asked to transcribe 91 different stimuli. These rhythms consisted of four onsets $t_0 \ldots t_3$, where t_0 and t_3 were fixed and occurred exactly on the beat (see Figure 10). First a pulse was provided to subjects, and then the stimulus was repeated three times with an empty bar between each repetition. Subjects were allowed to use any notation as a response and to listen to the stimulus as often as desired. In total, subjects used 125 different notations, of which 57 were used only once and 42 were used more than 3 times. An example is shown in Figure 11a. From this data, we estimate the posterior as

$$q(c_j \mid t_k) = n_k(\mathbf{c}_j) / \sum_{j} n_k(\mathbf{c}_j) \tag{18}$$

where $n_k(\mathbf{c}_j)$ denotes the number of times the stimulus $\mathbf{t}_k$ is associated with the notation $\mathbf{c}_j$.

Figure 10. Stimulus of the perception task (a) and the stimuli for the perception experiment (b). The dots denote the rhythms $\mathbf{t}_k$, where $k = 1...91$. Grid spacing is 56 msec.



Production Task

In the production task, the subjects were asked to perform the rhythms that they notated in the perception task. An example is shown in Figure 11b. For each notation $\mathbf{c}_j$ we assume a Gaussian distribution where

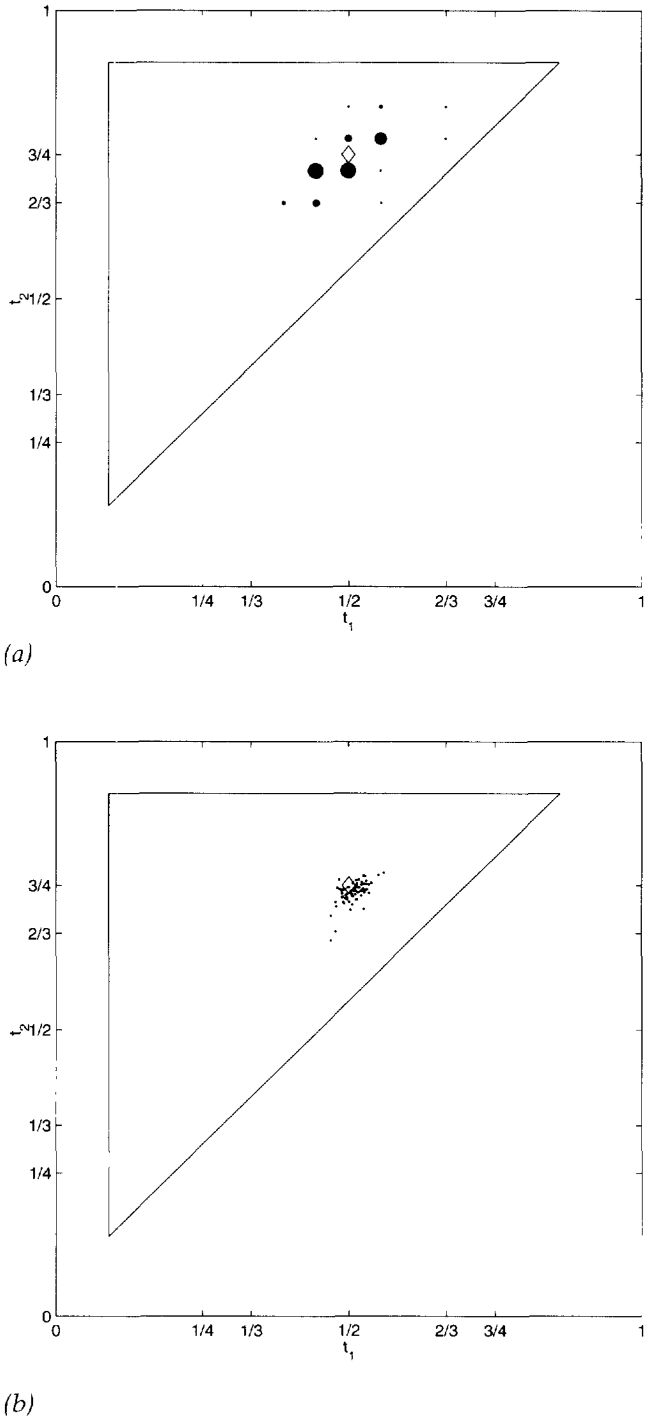
$$\hat{q}(\mathbf{t} \mid \mathbf{c}_j) = N(\mu_j, \Sigma_j). \tag{19}$$

The mean and the covariance matrix are estimated from production data by

$$\begin{aligned} \mu_j &= \frac{1}{N_j} \sum_k \mathbf{t}_{k,j} \\ \Sigma_j &= \frac{1}{N_j - 1} \sum_{k\,l} (\mathbf{t}_{k,j} - \mu_j)(\mathbf{t}_{l,j} - \mu_j)^T \end{aligned} \tag{20}$$

where $\mathbf{t}_{k,j}$ is the kth performance of $\mathbf{c}_j$ and N_j is the

Figure 11. Perception (a) and production (b) of the rhythm [2 1 1] ($\mathbf{c}$ = [0.5,0.75]). The diamond corresponds to the mechanical performance. In (a), the size of the circles is proportional to the estimated posterior $q(\mathbf{c}_j \mid \mathbf{t}_k)$. In (b), the dots correspond to performances of the rhythm.



total count of these performances in the data set. In the previous section, we proposed a model in which the correlation between two onsets decreases with increasing inter-onset interval. The correlation coefficient and the estimated error bars are depicted in Figure 12, where we observe that the correlation decreases with increasing distance between onsets.

Estimation of Model Parameters

The probabilistic model $p(\mathbf{c} \mid \mathbf{t})$ described in the previous section can be fitted by minimizing the distance to the estimated target $q(\mathbf{c} \mid \mathbf{t})$. A well-known distance measure between two probability distributions is the Kullback-Leiber (KL) divergence (Cover and Thomas 1991), given as

$$\mathrm{KL}(q \,\|\, p) = \int d\mathbf{x} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})}. \tag{21}$$

The integration is replaced by summation for discrete-probability distributions. It can be shown that $\mathrm{KL}(q \,\|\, p) \geq 0$ for any q,p and vanishes if and only if $q = p$ (Cover and Thomas 1991).

The KL divergence can be interpreted as a weighted average of the function $\log q(\mathbf{x})/p(\mathbf{x})$ with respect to weighting function $q(\mathbf{x})$. If $q(\mathbf{x})$ and $p(\mathbf{x})$ are significantly different for some $\mathbf{x}$ (for which $q(\mathbf{x})$ is sufficiently large), the KL divergence would also be large, and would indicate that the distributions are different. On the other hand, if the distributions have almost the same shape, $\log q(\mathbf{x})/p(\mathbf{x}) \approx 1$ for all $\mathbf{x}$ and the KL divergence would be close to zero since $\log(1) = 0$.

The KL divergence is an appropriate measure for the rhythm-quantization problem. We observed that for many stimuli, subjects gave different responses, and consequently it is difficult to choose just one "correct" notation for a particular stimulus. In other words, the mass of the target distribution $q(\mathbf{c} \mid \mathbf{t})$ is distributed among several code vectors. By minimizing the KL divergence, one can approximate the posterior distribution by preserving this intrinsic uncertainty. The optimization problem for the perception task can be set as

Table 2. Subdivisions

i	S_i
1	[2,2,2,2]
2	[3,2,2]
3	[3,3,2]
4	[5,2]
5	[7,2]
6	[11]
7	[13]
8	[5,3]
9	[17]
10	[7,3]

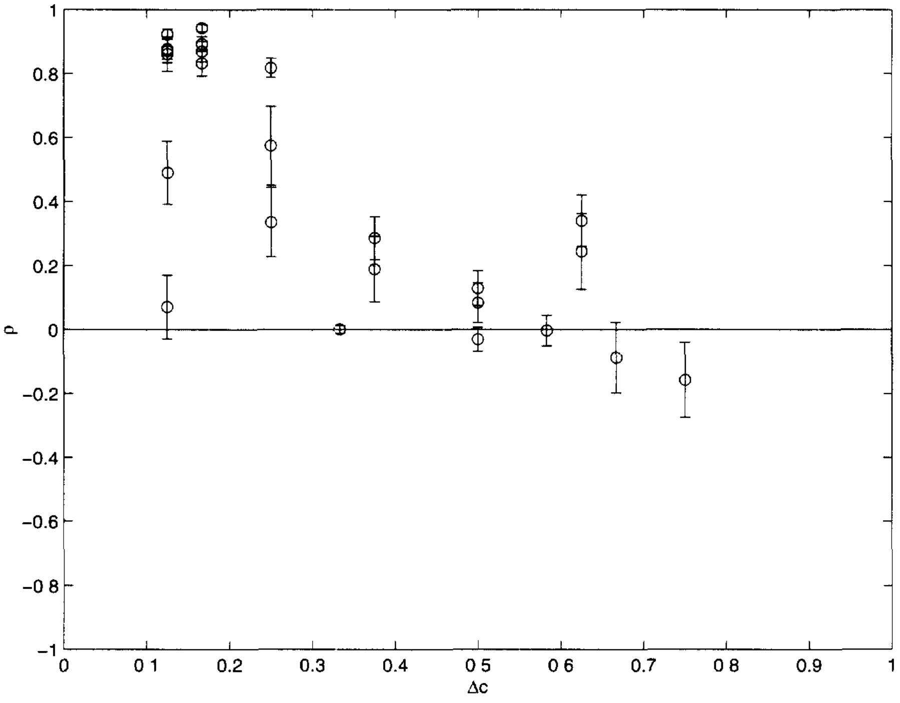
$$\begin{aligned} &\min \mathrm{KL}\left(q(\mathbf{c} \mid \mathbf{t})s(\mathbf{t}) \,\|\, p(\mathbf{c} \mid \mathbf{t})s(\mathbf{t})\right) \\ &\text{s.t. } \sigma > 0, -1 < \eta < 1, \\ &\lambda, \xi, \gamma \, \text{unconstrained} \end{aligned} \tag{22}$$

where $s(\mathbf{t}) \propto \sum_k \delta(\mathbf{t} - \mathbf{t}_k)$ is the distribution of the stimuli. This is a distribution that has positive mass only on the stimuli points $\mathbf{t}_k$. This measure forces the model to fit the estimated posterior at each stimulus point $\mathbf{t}_k$. We note that

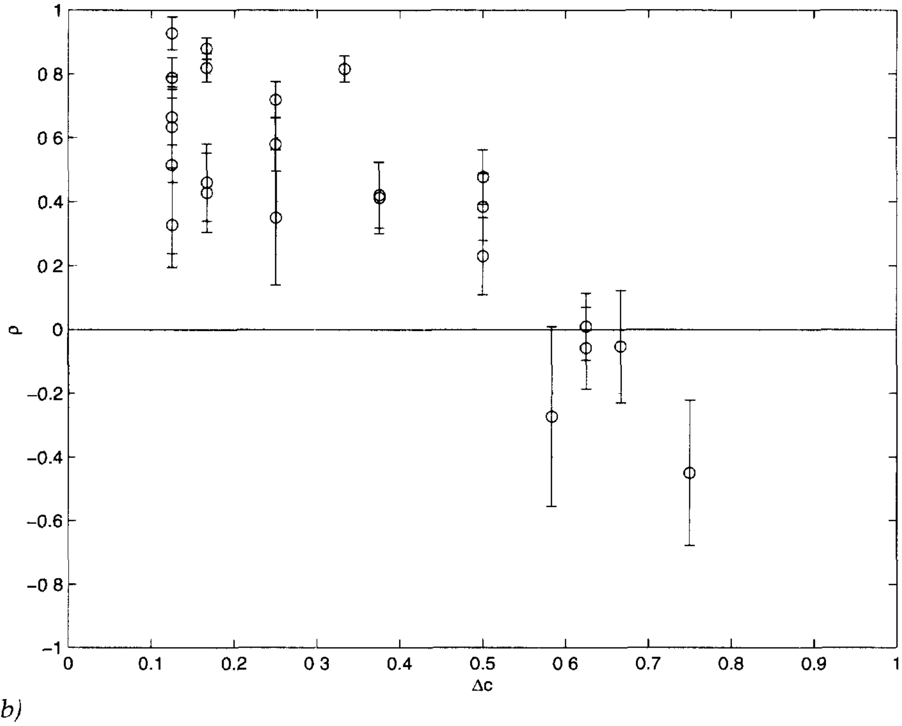
$$p(\mathbf{c} \mid \mathbf{t}) = \frac{p(t \mid c; \sigma, \lambda, \eta) p(c; \xi, \gamma)}{\sum_c p(t \mid c; \sigma, \lambda, \eta) p(c; \xi, \gamma)}. \tag{23}$$

This is in general a difficult optimization problem, owing to the presence of the denominator. Nevertheless, because the model has only five free parameters, we were able to minimize equation 22 by a standard BFGS quasi-Newton algorithm (Matlab function fminu). In our simulations, we observed that the objective function was rather smooth, and the optimum found was not sensitive to starting conditions, which suggests that there are not many local minima present.

Figure 12. Estimated correlation coefficient as a function of $\Delta c = c_2 - c_1$ on all subject responses: production (a); and perception (b).



(a)



(b)

Table 3. Optimization results, $\mathrm{CR}_{\mathrm{target}}$ = 48.0; = indicates values fixed during optimization; ! indicates values estimated from production data

Model		*Prior*		*Likelihood*		*Results*	
Label	ξ	γ	σ	λ	η	KL	$\mathrm{CR}_{\mathrm{model}}/\mathrm{CR}_{\mathrm{target}}$
I	1.35	0.75	0.083	2.57	0.66	1.30	77.1
II	1.34	0.75	0.086	=0	=0	1.41	71.3
III	1.33	0.77	0.409	=0	=0.98	1.96	51.4
IV	1.34	0.74	0.084	=0	0.39	1.34	75.3
V	=0	=0	0.085	=0	=0	1.92	29.7
VI	=0	=0	0.083	2.54	0.66	1.89	32.7
VII	1.43	0.79	!0.053	!3.07	!0.83	1.89	84.3

Results

The model was trained on a subset of the perception data by minimizing equation 22. In the training, we used 112 different notations (out of 125 that the subjects used) that could be generated by one of the subdivision schemas in Table 2. To identify the relative importance of model parameters, we optimized equation 22 by clamping some parameters.

We use a labeling of different models as follows: Model I is the complete model, where all parameters are unclamped. Model II is an onset quantizer ($\Sigma = \sigma^2 I$), where only prior parameters are active. Model III is (almost) an IOI quantizer, where the correlation between onsets is taken to be $\rho = 0.98$. Model IV is similar to Model I, with the simplification that the covariance matrix is constant for all code vectors. Since $\lambda = 0$ and $\rho = \eta$, Model V is an onset quantizer with a flat prior, similar to the quantizers used in commercial notation packages. Model VI has only the performance model parameters active.

In Model VII, the parameters of the performance model p ($\mathbf{t} \mid \mathbf{c}$) were estimated from the production data. The model was fitted to the production data $\hat{q}$ by minimizing

$$\mathrm{KL}\left(\hat{q}(\mathbf{c} \mid \mathbf{t})q(\mathbf{t}) \,\middle\|\, p(\mathbf{c} \mid \mathbf{t})q(\mathbf{t})\right) \tag{24}$$

where $q(\mathbf{c}_j) = \sum_k n_k(\mathbf{c}_j) / \sum_{k,j} n_k(\mathbf{c}_j)$, i.e., a histogram obtained by counting the subject responses in the perception experiment.

Although approximating the posterior at stimuli points was our objective in the optimization, for automatic transcription we were also interested in the classification performance. At each stimuli $\mathbf{t}_k$ by selecting the response that the subjects chose the most, i.e., $\mathbf{c}_k^* = \arg\max_c q(\mathbf{c} \mid \mathbf{t}_k)$, we could achieve the maximum possible classification rate on this data set, given as

$$\mathrm{CR}_{\mathrm{target}} \frac{n_k(\mathbf{c}_k^*)}{Z} \times 100. \tag{25}$$

Here, $Z = \sum_{k,\mathbf{c}} n_k(\mathbf{c}_k^*)$, the total number of measurements. Similarly, if we select the code vector with the highest predicted posterior $\mathbf{c}_k^* = \arg\max_{\mathbf{c}} p(\mathbf{c} \mid \mathbf{t}_k)$ at each stimulus, we achieve the classification rate of the model denoted as $\mathrm{CR}_{\mathrm{model}}$. The results are shown in Table 3. The clamped parameters are tagged with an "equal" sign. The results shown are for a code book consisting of 112 code vectors, which the subjects used in their responses and could have been generated by one of the subdivisions in Table 2.

Model I performs the best in terms of the KL divergence. However, the marginal benefit obtained by choosing a correlation structure—which decreases with increasing onset distances (obtained by varying λ)—is rather small. One can achieve almost the same performance by having a constant correlation between onsets (Model IV). By comparing Model IV to Models II and III, we can say that, under the given prior distribution, the subjects are employing a quantization strategy that is somehow between pure onset quantization and IOI quantization. The choice of the prior is important, which can be seen from the results of Models V and VI, which perform poorly due to the flat prior assumption.

Model VII suggests that for this data set (under the assumption that our model is correct) the perception and production processes are different. This is mainly owing to the spread parameter T, which is smaller for the production data. The interpretation of this behavior is that subjects deviate less from the mechanical mean in a performance situation. However, this might be because performances were carried out devoid of any context, thereby forcing the subjects to concentrate on exact timing. It is interesting to note that almost the same correlation structure is preserved in both experiments. This suggests that there is some relationship between the production and perception process. The classification performance of Model VII is surprisingly high; it predicts the winner accurately. However, the fit of the posterior is poor, which can be seen by the high KL-divergence score.

For visualization of the results, we employ an interpolation procedure to estimate the target posterior at other points than the stimuli (see the appendix). The rhythm space can be tiled into regions of rhythms that are quantized to the same code vector. Estimated tiles from experimental data are given in Figure 13a.

In practice, it is not feasible to identify explicitly a subset of all possible code vectors that have nonzero prior probability. For example, the number of notations that can be generated by subdivisions in Table 2 is 886, whereas the subjects used only 112 of these as responses. This subset must be predicted by the model as well. A simple grid quantizer tries to approximate this subset by assigning a constant prior probability to code vectors only up to a certain threshold depth. The proposed prior model can be contrasted to this schema in that it distributes the probability mass in a perceptually more realistic manner. To visualize this, we generated a code book consisting of all 886 code vectors. The tilings generated by Model I and Model V for this code book are depicted in Figures 13b and 13c. To compare the tilings, we estimate the ratio

$$\text{Match} = \frac{A_{\text{match}}}{A_{\text{total}}} \times 100 \qquad (26)$$

where A_{match} is the area where the model matches with the target, and A_{total} is the total area of the triangle. Note that this is just a crude approximation to the classification performance under the assumption that all rhythms are equally probable. The results are shown in Table 4.

Discussion and Conclusion

In this article, we developed a vector quantizer for transcription of musical performances. We considered the problem in the framework of Bayesian statistics, where we proposed a quantizer model. Experimentally, we observe that even for quantization of simple rhythms, well-trained subjects give quite different answers. Clearly, in many cases, there exists more than one correct notation. In this respect, probabilistic modeling provides a natural framework.

The quantizer depends upon two probability models: a performance model, and a prior. The performance model generalizes simple quantization strategies by taking the correlation structure of the music into account; for example, onset quantization appears as a special case. The particular parametric form is shown to be perceptually meaningful, and facilitates efficient implementation. It can also be interpreted as a suitable distance measure between rhythms.

The prior model can be interpreted as a complexity measure. In contrast to the likelihood, which has a rather standard form, the prior reflects our intuitive and subjective notion about the complexity of a notation, and derives from consideration of time signatures and the hierarchical (i.e., tree-like) structure of musical rhythms. The model

Figure 13. Tilings of the rhythm space: target (a); Model I, (ξ, γ, σλ, η) = (1.25,0.75,.083,2.57,0.66) (b); and Model V, (ξ, γ, σ, λ, η) = (0,0,0.085,0,0). c). The tiles denote the sets of rhythms, which would be quantized to the same code vector. Both Models I and V use the same code book of 886 code vectors. Since Model V assigns the same prior probability to all code vectors, the best code vector is always the nearest code vector (in Euclidian distance), and consequently, the rhythm space is highly fragmented.

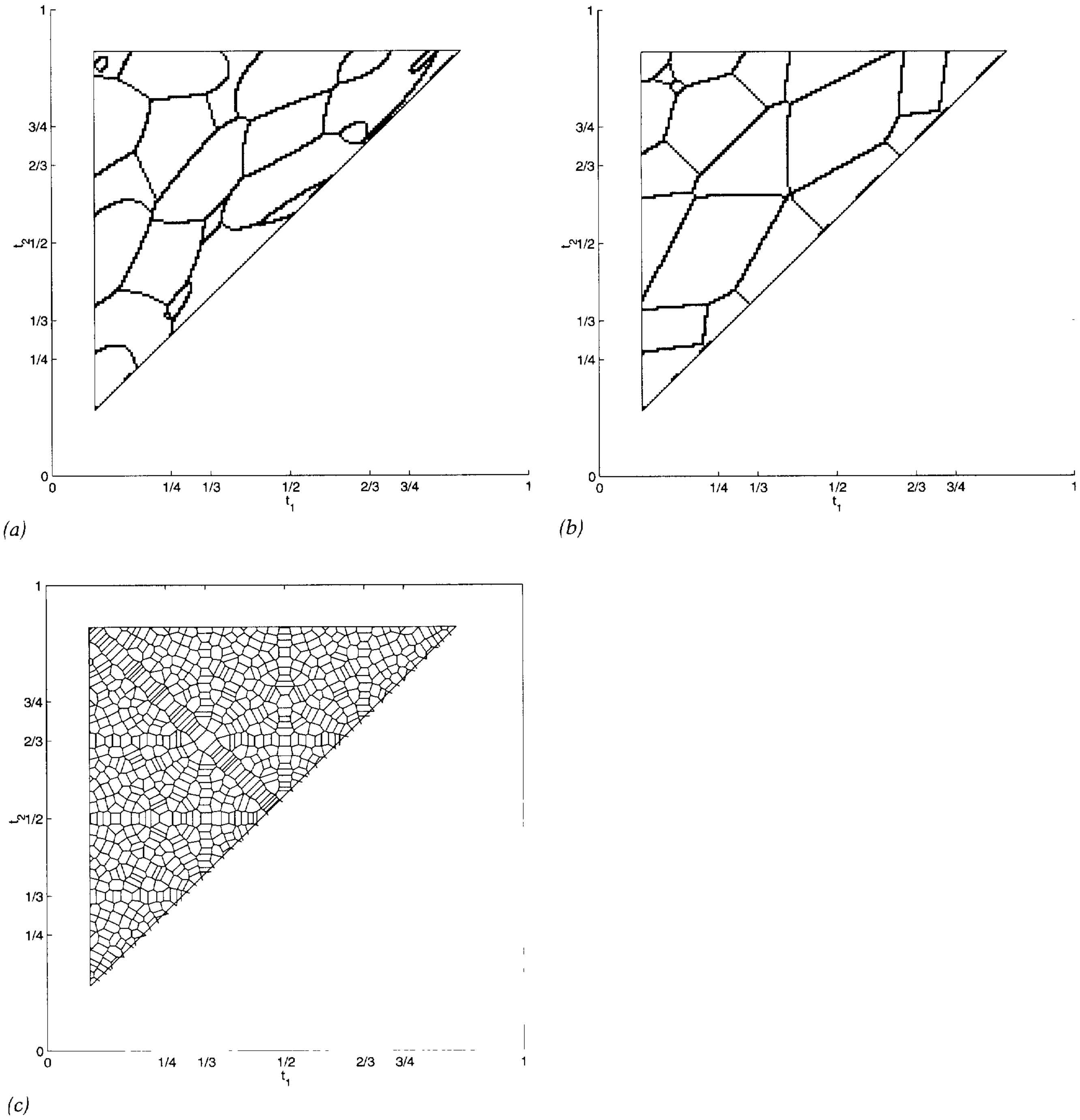


Table 4. Amount of match between tilings generated by the target and models

	I	*II*	*III*	*IV*	*V*	*VI*	*VII*
Match	58.8	53.5	36.1	59.0	3.8	3.1	56.7

is verified and optimized by data obtained from a psychoacoustical experiment. The optimization results suggest that prior and likelihood parameters can be optimized independently, since clampıng one set of parameters affects the optimal values of others only very slightly. This property makes the interpretation of the model easier. Since we explicitly state the probability model, we can make comparisons between models by using the KL divergence as a goodness of fit measure. Indeed, any other model that computes a posterior distribution $p(\mathbf{c} \mid \mathbf{t})$ could be compared in a quantitative manner using this framework. A class of statistical tests to determine whether one model is significantly better than another is known as *bootstrapping* (Efron and Tibshirani 1993). This methods can be used to estimate error bars on the KL-divergence measures to determine any significant differences between models.

We must stress the point that the particular parameter settings we found from our data do not represent the ultimate means of performing quantization for every circumstance. First, the model does not use any other attributes of notes (e.g., duration or pitch), which might provide additional information and hence a better quantization. Second, we have not addressed the context information. Theoretically, such improvements could be integrated by proposing more complex likelihood and prior models. As already demonstrated, since all the assumptions are stated as distributions, corresponding optimal parameters can be estimated from experimental data. A practical but important limitation is that parameter estimation in more complex models requires a larger data set; otherwise, the estimation can be subject to overfitting. A large data set is difficult to collect, since one effectively must rely on psychoacoustical experiments, which are inherently limited in the number of experimental conditions one can impose (e.g., number of onsets, tempo, context, etc.). Nevertheless, we believe that the current framework is a consistent and principled way to investigate the quantization problem.

Acknowledgments

This research is supported by the Technology Foundation STW, applied science division of NWO, and the technology program of the Dutch Ministry of Economic Affairs. The first author is thankful to David Barber for stimulating discussions.

References

Agon, C., G. Assayag, J. Fineberg, and C. Rueda. 1994. "Kant: A Critique of Pure Quantification." *Proceedıngs of the International Computer Music Conference.* San Francısco: International Computer Music Association, pp. 52–59.

Clarke, E. F. 1985. "Structure and Expression in Rhythmic Performance." In P. Howell, I. Cross, and R. West, eds. *Musical Structure and Cognition.* London: Academic Press.

Cover, T. M., and J. A. Thomas. 1991. *Elements of Information Theory.* New York: Wiley.

Desain, P., R. Aarts, A. T. Cemgil, B. Kappen, H. van Thienen, and P. Trilsbeek. 1999. "Robust Time-Quantization for Music." Preprint 4905-H4 of Audio Engineering Society (AES) 106th Conventıon. Munich, Germany: AES.

Desain, P., and H. Honing. 1991. "Quantization of Musical Time: A Connectionist Approach." In P. M. Todd and D. G. Loy, eds. *Music and Connectionism.* Cambridge, Massachusetts: MIT Press, pp. 150–167.

Desain, P., H. Honing, and K. de Rijk. 1992. "The Quantization of Musıcal Time: A Connectionist Approach." In *Music, Mınd, and Machıne: Studies in Computer Musıc, Music Cognitıon, and Artificial Intelligence.* Amsterdam: Thesis Publishers, pp. 59–78.

Duda, R. O., and P. E. Hart. 1973. *Pattern Classification and Scene Analysis.* New York: Wiley.

Efron, B., and R. Tibshirani. 1993. *An Introduction to the Bootstrap.* New York: Chapman and Hall.

Large, E. W. 1995. "Beat Tracking with a Nonlinear Oscillator." In *Working Notes of the IJCAI Workshop on AI and Music: International Joint Conferences on Artificial Intelligence.* Montreal, pp. 24–31.

Longuet-Higgins, H. C. 1987. *Mental Processes: Studies in Cognitive Science.* Cambridge, Massachusetts: MIT Press, p. 424.

Pressing, J., and P. Lawrence. 1993. "Transcribe: A Comprehensive Autotranscription Program." *Proceedings of the International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 343–345.

Toiviainen, P. 1999. "An Interactive MIDI Accompanist." *Computer Music Journal* 22(4):63–75.

Appendix: Estimation of the Posterior from Subject Responses

Let $\mathbf{t}_k$ be the stimuli points. The histogram estimate at $\mathbf{t}_k$ is denoted by $q(\mathrm{c}_l \mid \mathbf{t}_k)$, We define a kernel

$$G(\mathbf{t};\mathbf{t}_0,\sigma) = \exp(-\frac{\left\|\mathbf{t}-\mathbf{t}_0\right\|^2}{2\sigma^2} \qquad (27)$$

where $\|\mathbf{x}\|$ is the length of the vector $\mathbf{x}$. Then the posterior probability of $\mathbf{c}_l$ at an arbitrary point $\mathbf{t}$ is given as

$$q(\mathbf{c}_l \mid \mathbf{t}) = \sum_k \alpha_k(\mathbf{t}) q(\mathbf{c}_l \mid \mathbf{t}_k) \qquad (28)$$

where $\alpha_k(\mathbf{t}) = \frac{G(t;t_k,\sigma}{\sum_r G(t;t_r,\sigma)}$. We have taken $\sigma = 0.04$.