Rhythm Quantization for Transcription

Ali Taylan Cemgil*; Peter Desain†; Bert Kappen*
*SNN, University of Nijmegen, The Netherlands
†NICI, University of Nijmegen, The Netherlands
cemgil@mbfys.kun.nl

Abstract

Automatic Music Transcription is the extraction of an acceptable notation from performed music. One important task in this problem is rhythm quantization which refers to categorization of note durations. Although quantization of a pure mechanical performance is rather straightforward, the task becomes increasingly difficult in presence of musical expression, i.e. systematic variations in timing of notes and tempo changes. For quantization of natural performances, we employ a framework based on Bayesian statistics. Expressive deviations are modelled by a probabilistic performance model from which the corresponding optimal quantizer can be derived by Bayes theorem. We demonstrate that some simple quantization schemata can be derived in this framework by simple assumptions about timing deviations. A general quantization method, which can be derived in this framework, is vector quantization (VQ). The algorithm operates on short groups of onsets and is thus flexible in capturing the structure of timing deviations between neighbouring onsets and thus performs better than simple rounding methods. Finally, we present some results on simple examples.

1 Introduction

Automatic Music Transcription is the extraction of an acceptable musical description from performed music. The interest into this problem is motivated by the desire to design a program which creates automatically a notation from a performance. In general, e.g. when directly operating on an acoustical recording of polyphonic music (polyphonic pitch tracking), this task proved to be a very difficult one and stays yet as an unsolved engineering problem. Surprisingly, even a virtually 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 presence of expressive variations which can be thought as systematic deviations from a pure mechanical performance. In such unconstrained performance conditions, mainly two types systematic deviations from exact values do occur. In the small scale notes can be played accented or delayed. In the large scale tempo can vary, for example the musician(s) 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) which takes all possible background factors into account, seems to be quite hard. On the other hand if the quantizer does not incorporate any model for these deviations, the results are usually unsatisfactory. 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 made of onsets. This also suggests that attributes of neighbouring onsets such as duration, timing deviation etc. are correlated in some way. A good quantization schema should and must exploit this correlation structure. This structure is not fully exploited in commercial music packages which do automated music transcription and score type setting. 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 pre-specified minimum note duration (e.g. eight, sixteenth e.t.c.). Such a grid quantization schema implies that each onset is quantized to the nearest grid point *independent* of its neighbours and thus all of its attributes are assumed to be independent, hence the correlation structure is not employed. The consequence of this restriction is that users are restricted to play along with a fixed metronome and without any expression. From another point of view, the musician has to fit her performance to the performance model of the program. 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 indirectly take the correlation structure of expressive deviations into account. In one of the first attempt to quantization, Longuet-Higgins (1987) described a method which uses hierarchical structure of musical rhythms to do quantization. Desain et al. (1992) use a relaxation network in which pairs of time intervals are attracted to simple integer ratios. Pressing and Lawrence (1993) use several template grids and compare both onsets and inter onset intervals (IOI's) to the grid and select the best quantization according to some distance criterion. The Kant system Agon et al. (1994) developed at IRCAM uses more sophisticated heuristics but is in principle similar to (Pressing and Lawrence, 1993).

The common critic to all of these models is that the assumptions about the expressive deviations are implicit and are usually hidden in the model, thus it is not always clear how a particular design choice effects the overall performance for a full range of musical styles. In this paper we describe a framework which makes the assumptions more explicit. In the following section we state the transcription problem along with some simple examples. We give a brief summary of Bayesian statistics and graphical models. Using this framework, we describe several performance models and show how different quantizers can be derived from them. Finally, we compare the results.

2 Problem Description

We defined automated music transcription as the extraction of an *acceptable* description (music notation) from *performed* data. In this study we concentrate to a simplified problem, namely extraction of an acceptable notation for a performed simple rhythm (e.g. tapped by a pen). 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 suitable distances between onset times of the performed rhythm and the mechanical performance (e.g. as would be played by a computer). Consider an example from Desain and Honing (1991). We are given a segment of a performed rhythm as in Figure 1. A simple grid quantizer may produce the result in Figure 2.(a). Although this is a very accurate representation, musicians would probably agree that the "smoother" score shown in Figure 2.(b) is a better representation.

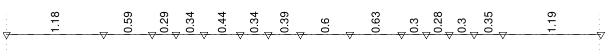

Figure 1: Example: A performed rhythm

This 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.

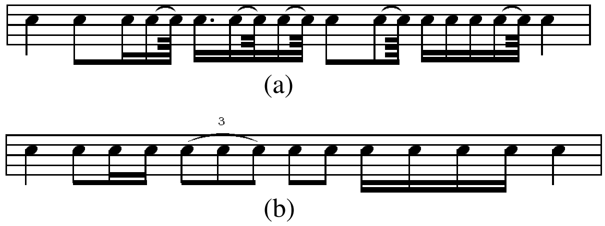

Figure 2: (a) "Too" accurate quantization (b) Desired quantization

3 Bayesian Statistics

An approach in solving difficult real world problems which require artificial intelligence techniques, involves to devise a model of the phenomenon, discuss its assumptions, and try to make predictions to test the its validity and applicability. This is a rather incremental process, in which the model is refined with the increasing domain knowledge. This domain knowledge can be roughly classified as experimental evidence (collected data) and expert knowledge (designers intuitions). One difficulty appears in the formal representation and application of expert knowledge, since it is usually in terms of a verbal description which can be prone to misinterpretation or can be hard to validate computationally. The other difficulty is in getting useful information out of the collected data. For this task, several different pattern recognition techniques (such as Neural Networks) can be used, which can capture the underlying structure of the phenomenon with quite general computational models, provided that enough experimental data is available. However, in the design of a working real world system one would like to combine expert knowledge with experimental evidence and Bayesian statistics provides a consistent and practical alternative for this requirement.

3.1 Bayes Theorem

The joint probability $p(A, B)$ of two discrete random variables A and B defined over the respective state spaces $\{a_1 \dots a_N\}$ and $\{b_1 \dots b_M\}$ can be factorized in two ways:

$$p(A, B) = p(B|A)p(A) = p(A|B)p(B) \tag{1}$$

where $p(A|B)$ denotes the conditional probability of A given B.

The marginal distribution of a variable can be found from the joint by summing over all states of the other variable, e.g.:

$$p(A) = \sum_i p(A, B = b_i) = \sum_i p(A|B = b_i)p(B = b_i) \tag{2}$$

It is understood that summation is to be replaced by integration if the state space is continuous.

Bayes theorem appears from Eq. 1 and Eq. 2:

$$p(B|A) = \frac{p(A|B)p(B)}{\sum_i p(A|B = b_i)p(B = b_i)} \tag{3}$$

This rather simple looking "formula" has surprisingly far reaching consequences. One reason is the interpretation of Eq. 3 as:

$$p(Model|Data) \propto p(Data|Model)p(Model) \quad (4)$$
$$\text{posterior} \propto \text{likelihood} \times \text{prior} \quad (5)$$

which combines "the amount of fit" (the likelihood) with the initial "subjective belief" (prior) to give the new (posterior) belief into the model after we see the data. In particular, if the model is an element of a class of models indexed by some parameters finding the most probable model is equivalent to finding the most probable set of parameters.

4 Rhythm Quantization Problem

4.1 Definitions

In this section we will give formal definitions of the terms that we will use in derivations to follow. A *performed rhythm* is denoted by $\mathbf{t} = [t_1, \ldots, t_i, \ldots t_I]$[1] where t_i is the time of occurrence of the i'th onset (measured in seconds). The time between two consecutive onsets is denoted as an inter onset interval (IOI) which is defined as $\delta_i = t_i - t_{i-1}$. For example, the rhythm in Figure 1 is represented by $\delta_1 = 1.18, \delta_2 = 0.59, \delta_3 = 0.29\ldots$, and $t_1 = 0, t_2 = 1.18, t_2 = 1.77, t_3 = 2.06\ldots$.

We found it convenient to make a distinction between a score and a performance, although a score corresponds directly to a mechanical performance when played with a constant tempo. We define a *score* $\mathbf{s} = [s_i], i = 1 \ldots N$ as a sequence of occurrence times of onsets as a multiple of some basic unit (e.g quarter note, eight note etc.).

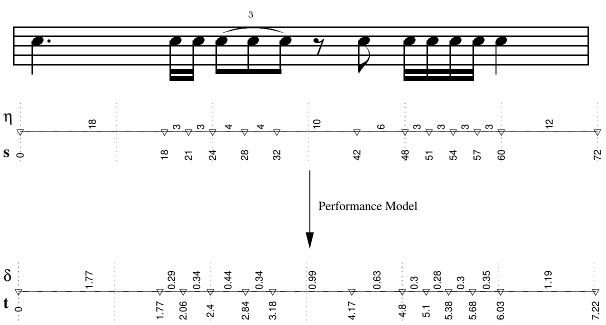

Figure 3: Representation

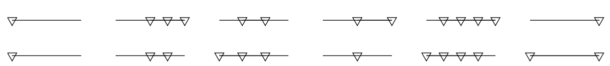

Figure 4: Two equivalent representations of the notation in Figure 3 by a code vector sequence

[1] We will denote a set with the typical element x_j as $\{x_j\}$. If the elements are ordered (e.g. to form a string) we will use $[x_j]$.

A score $\mathbf{s}$ can also be viewed as a *concatenation* of some basic building blocks $\mathbf{c}_j = [c_{1,j}, c_{2,j} \ldots c_{K,j}]$, which we call *code vectors*. For example, the notation in Figure 3 can be represented by a code vector sequence as in Figure 4. Note that the representation is not unique, both code vector sequences represent the same notation. We call a set of code-vectors a *codebook*.

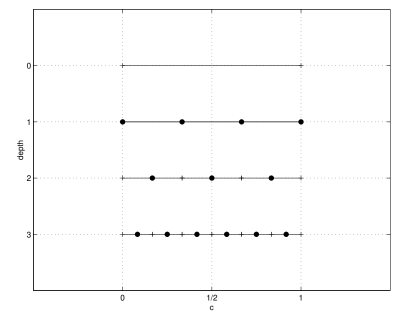

Figure 5: Depth of c by subdivision schema $\mathcal{S} = [3, 2, 2]$

In music, notations are usually generated by regular subdivisions of a time interval so the locations of onsets (in a score) can be described by simple rational numbers. For this reason, it is also convenient to generate code-vectors by regular subdivisions. This subdivision schema is usually related to the time signature of the bar. As an example, consider a codebook which contains only one onset code-vectors generated according to a 3/4 time signature. A possible subdivision schema could be $\mathcal{S} = [3, 2, 2]$. The interval is divided first to 3, then resulting three intervals into 2 and etc. At each iteration, the end-points of the generated intervals, which are not already in the codebook are added to the codebook. The resulting codebook is depicted in Figure 5. The filled circles correspond to the code-vectors and are arranged by the location of the onset (horizontal) and by their *depth* (vertical). The depth of an onset (with respect to a subdivision schema $\mathcal{S} = [n_i]$) is the index of the iteration at which it is added to a codebook as $d(c|\mathcal{S})$. A code-vector with N onsets can be build by combining of such one onset code-vectors. For such a code-vector we define the depth as the sum of the depths of its onsets:

$$d(\mathbf{c}|\mathcal{S}) = \sum_{c_n \in \mathbf{c}} d(c_n|\mathcal{S}) \quad (6)$$

For simplification of notation, we will represent a performance $\mathbf{t}$ as a sequence of non-overlapping segments. The length of the j'th segment is denoted as Δ_j and the onsets in this segment are denoted as $\mathbf{t}_j$. The segmentation of a performance is given in Figure 6. The onsets are normalized by Δ_j so an onset at the beginning of the segment is mapped to zero and one at the end to one.

4.2 Performance Model

In general terms, a *performance model* describes how a score is mapped into a performance. (Figure 3). As de-

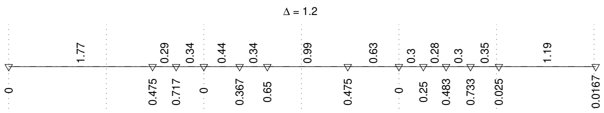

Figure 6: Segmentation of a performance by constant Δ.

scribed in the introduction section, natural musical performance is subject to several systematic deviations. In lack 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 performance of very short rhythms show deviations from a strict mechanical performance and a performance model is a description of such deviations. In this section we describe a probabilistic model which can capture some of the properties of expressive timing deviations.

In the probabilistic framework, a performance model is the probability distribution $p(\mathbf{t}, \mathbf{c}, \mathbf{v}, \mathcal{P})$, where $\mathbf{c}$ is a code-vector sequence corresponding to a score $\mathbf{s}$ and $\mathbf{v} = [v_j] > 0\, \forall j$ is a *tempo curve* (measured in beats/sec). If the tempo v_j is constant we drop the index j and use v or its reciprocal $\Delta = 1/v$. $\mathcal{P}$ is a set of background parameters which can include several aspects such as style, instrument etc. In our subsequent derivations we will simply ignore it. A performed rhythm $\mathbf{t}$ is viewed as a single realization from a performance model.

The form of a performance model can be in general very complicated. However, in this article we will consider a subclass of performance models where the the expressive timing is assumed to be an additive noise component. The model is given by

$$\mathbf{t}_j = \frac{\mathbf{c}_j}{v_j} + \varepsilon_j \tag{7}$$

where ε_j is a random variable denoting the *expressive timing deviation*, and $v_j > 0$ is a *tempo curve*. Note that when ε_j is zero and v is constant, the model reduces to a so-called "mechanical" performance. Although this simple additivity assumption can be restrictive, this model is still quite general since we haven't yet made any assumptions about the dependence between tempo, code-vectors and expressive timing deviations.

4.3 Formal statement of the Rhythm Quantization Problem

Given a performed rhythm $\mathbf{t}$, (or equivalently corresponding IOI's δ) find a code-vector sequence $\mathbf{c}^*$ and a tempo curve $\mathbf{v}^*$ s.t. $p(\mathbf{c}, \mathbf{v}|\mathbf{t}, \mathcal{P})$ is maximized, i.e. :

$$(\mathbf{c}^*, \mathbf{v}^*) = \operatorname*{argmax}_{\mathbf{c},\mathbf{v}} p(\mathbf{c}, \mathbf{v}|\mathbf{t}, \mathcal{P}) \tag{8}$$

This problem can be viewed as a maximum a-posteriori (MAP) estimation problem if we regard $\mathbf{c}$ and $\mathbf{v}$ as parameters of the performance model and determine the values which maximize a-posteriori probability given in Eq. 9.

$$\begin{aligned} p(\mathbf{c}, \mathbf{v}|\mathbf{t}, \mathcal{P}) &\propto p(\mathbf{t}|\mathbf{c}, \mathbf{v}, \mathcal{P}) p(\mathbf{c}, \mathbf{v}|\mathcal{P}) && (9) \\ \text{Score and Tempo} &\propto \text{Performance Likelihood} \\ &\quad \times \text{Score and Tempo Prior} && (10) \end{aligned}$$

We can also define a related quantity $\mathcal{L}$ (minus log-posterior) and try to minimize this quantity rather then maximizing Eq. 9 directly. This simplifies the form of the objective function without changing the locations of local extrema since $\log(x)$ is a monotonicly increasing function.

$$\mathcal{L} = -\log p(\mathbf{c}, \mathbf{v}|\mathbf{t}) = -\log p(\mathbf{t}|\mathbf{c}, \mathbf{v}) - \log p(\mathbf{c}, \mathbf{v}) \tag{11}$$

4.4 Derivation of a Vector Quantizer with constant Tempo

We will now demonstrate the derivation of a quantizer using the performance model in Eq. 7 but with constant tempo v:

$$\mathbf{t}_j = \mathbf{c}_j / v + \varepsilon_j \tag{12}$$

Since the tempo is constant, we have to specify only a scalar v as the tempo, i.e. we have to specify the tempo prior $p(v)$ which should reflect our preferences toward slower tempos, or equivalently to longer beat lengths. Otherwise any rhythm can be notated by a simple score (e.g. tied whole notes) if the tempo is very fast. Clearly, there must be a penalty term to avoid this undesired situation. A reasonable choice for the prior seems to be the exponential distribution $p(v) = \lambda e^{-\lambda v}$ where λ is the expected grid length, which determines the rate of decay of the exponential. If λ is big, the exponential approaches zero quickly and the probability mass assigned to faster v decreases.

The code-vectors are assumed to be independent, i.e. $p(\mathbf{c}_j, \mathbf{c}_k) = p(\mathbf{c}_j)p(\mathbf{c}_k)$. The expressive noise component is assumed to be Normal distributed with zero mean and covariance matrix Σ_ε, i.e. $p(\varepsilon) = \mathcal{N}(0, \Sigma_\varepsilon)$.

If we substitute these assumptions to Eq. 11 we get

$$\begin{aligned} \mathcal{L} &= -\sum_j \{\log p(\mathbf{t}_j|\mathbf{c}_j, v) + \log p(\mathbf{c}_j)\} - J \log p(v) && (13) \\ &= \sum_j \frac{1}{v^2} (\mathbf{t}_j v - \mathbf{c}_j)^T \Sigma_\varepsilon^{-1} (\mathbf{t}_j v - \mathbf{c}_j) \\ &\quad + \sum_j \log \frac{1}{p(\mathbf{c}_j)} + J\lambda v && (14) \\ &= \text{Quantization Error} \\ &\quad + \text{Score Complexity} + \text{Tempo Penalty} && (15) \end{aligned}$$

The first term in Equation 14 is the square of a weighted Euclidian distance (Mahalanobis distance (Duda and Hart, 1973)) measuring how far the rhythm is played from the perfect mechanical performance. The covariance matrix

Σ_ε is determined by the correlation structure of the expressive noise and will be explained in the next section. The second term, which is large when the prior probability $p(\mathbf{c}_j)$ of the codevector is low, is the length of a Shannon code in bits (Cover and Thomas, 1991). This term can be interpreted as a complexity term, which penalizes complex notations. Finally, the third term prefers slower tempos. The best quantization balances these three terms in an optimal way.

4.5 A Special case: Regularized Grid Quantizer

The equation 14 can be shown to reduce to a grid quantizer under the assumption that expressive deviations of individual onsets are independent, i.e. $\Sigma_\varepsilon = \sigma_\varepsilon^2 \mathbf{I}$ if the codebook is taken to be $\mathbf{C} = \{[0], [1], [\emptyset]\}$.

$$\mathcal{L}/J = E(v, \mathbf{c}) = \frac{1}{2J\sigma_\varepsilon^2} \sum_j \frac{(v\mathbf{t}_j - \mathbf{c}_j)^2}{v^2} + \lambda v + const \quad (16)$$

$$\mathbf{c}^*{}_j = \mathrm{round}(\mathbf{t}_j v) \quad (17)$$

4.6 Vector Quantizer

In this section we will demonstrate the advantages of quantization of onsets in groups (vector quantization) and the flexibility of this schema by a simple example. To simplify the derivation further, we will only consider the quantization of a single segment $\mathbf{t}_j$.

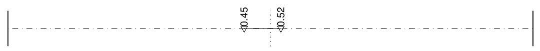

Figure 7: Two Onsets

Consider the normalized segment $\mathbf{t}_j v_j = [0.45, 0.52]$ depicted in Figure 7. Suppose we wish to quantize the onsets only to one of the endpoints, i.e. we are using effectively the codebook $\mathbf{C} = \{[0,0], [0,1], [1,1]\}$. 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 very desirable, since the IOI has increased more than 14 times, (from 0.07 to 1). This is less likely since it is perceptually not very realistic. We could fix this problem by employing another strategy : If $\delta > 0.5$, we use the code-vector $[0, 1]$. if $\delta = t_2 - t_1 < 0.5$, we quantize to one of the code-vectors $[0, 0]$ or $[1, 1]$ depending upon the average of the onsets. In this strategy the quantization of $[0.45, 0.52]$ is $[0, 1]$.

Although considered to be different in the literature, both strategies are just special cases which can be derived from the equation 14 by making specific choices about the complexity prior and the correlation structure (covariance matrix Σ_ε) of expressive deviations. The first strategy ($\rho = 0$) assumes that the expressive deviations are independent of each other and distributed by $\mathcal{N}(0, \sigma^2)$.

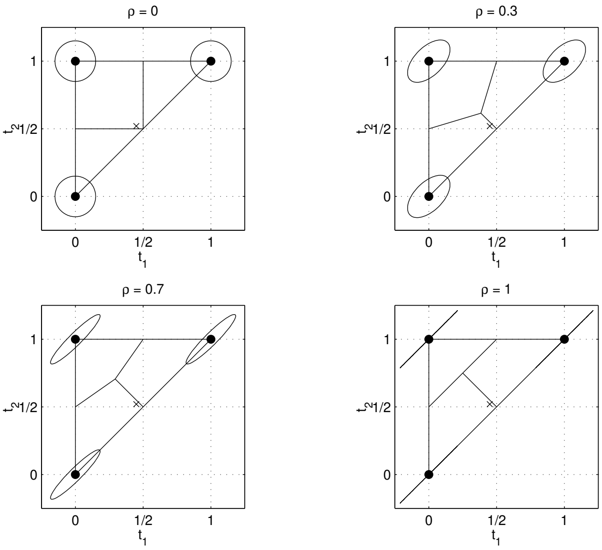

Figure 8: Tilings for choices of ρ and constant $p(\mathbf{c})$

The latter corresponds to the case where onsets are assumed to have a linear functional dependence; which is indeed true since this strategy considered that $t_2 = t_1 + \delta$. Different strategies, which can be quite difficult to state verbally, can be specified by different choices of Σ_ε and $p(\mathbf{c})$. Some examples for the choice $\Sigma_\varepsilon = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ with constant $p(\mathbf{c})$ are depicted in Figure 8. The ellipses, who's orientation is determined by the covariance matrix, denote the set of points which are equidistant from the center. The interested reader is referred to Duda and Hart (1973) for a detailed discussion of the underlying theory.

4.6.1 Likelihood and Prior for the Vector Quantizer

To choose the likelihood $p(t|c, v)$ and the prior $p(c)$ in a way which is perceptually meaningful, we analysed data from an experiment where subjects are asked two notate short rhythms Desain et al. (1999).

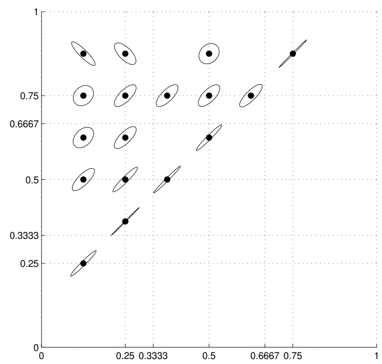

Figure 9: Correlations estimated from the perceptual experiment

Analysis of data suggests that onsets which 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 difference is large, the correlation tends to be weak (or sometimes negative), which suggests that onset are quantized independently of each other with respect to the tactus (i.e. the grid) only.

By this observation we found it useful to define the covariance matrix as:

$$\Sigma_\varepsilon = (\sigma^2_{mn}) \qquad (18)$$
$$\sigma^2_{mn} = f(v|t_m - t_n|) \qquad (19)$$

where f is a monotonically decreasing function where $f(0) = 1$ and $f(1) = 0$. We have choosen $f(x) = (1-x)^\alpha$ where $\alpha > 0$. In practice it is taken around 1.

The choice of the prior $p(\mathbf{c})$ reflects the complexity of $\mathbf{c}$. We think that the complexity can be related to the number of subdivisions required to encode the onset. Another factor in the determination of the complexity of code-vectors is how the onsets of a code-vector are distributed on the interval. As an example consider two code-vectors $\mathbf{c}_1 = [0, 1/2, 3/4, 1]$ and $\mathbf{c}_2 = [0, 1/4, 3/4, 1]$ which correspond to notations (211) and (121). We consider the latter to be more complex since it contains more onsets which are located on “deeper” locations.

The prior probability of a code-vector with respect to $\mathcal{S}$ is choosen as

$$p(\mathbf{c}|\mathcal{S}) = \frac{1}{Z}\gamma^{-d(\mathbf{c}|\mathcal{S})} \qquad (20)$$

where Z is a normalizing constant. Note that if $\gamma = 1$, then the depth of the codevector has no influence upon its complexity. If it is large, (e.g. $\gamma = 2$) only very simple rhythms get reasonable probability mass. In practice, we choose $\gamma \approx 1.02$. This choice is also in accordance with the intuition and experimental evidence that simpler rhythms are more frequently used then complex rhythms.

5 Simulations

We implemented the grid quantizer with regularized tempo detector and tested it on the example in Figure 1. The regularized error is ploted in 10. The form of the error curves demonstrates the effect of the prior and how quantization error and tempo penalty are balanced. In any case a grid length Δ is found which is a suitable subdivision of the onset sequence. This is demonstrated in Figure 11 where the optimal grid length estimate as a function of the expected grid length λ is ploted. Note that the values Δ =0.15 0.3 and 0.4 correspond to 1/32, 1/16, and 1/12 notes. By varying λ, one can balance accuracy against complexity. Varying this parameter is similar to selecting the grid size of a grid quantizer but on a continuous

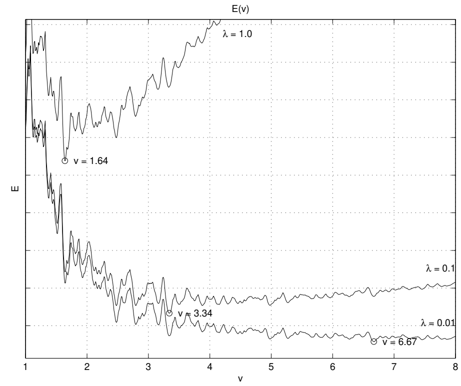

Figure 10: Regularized Error $\mathcal{L} \propto E(v)$

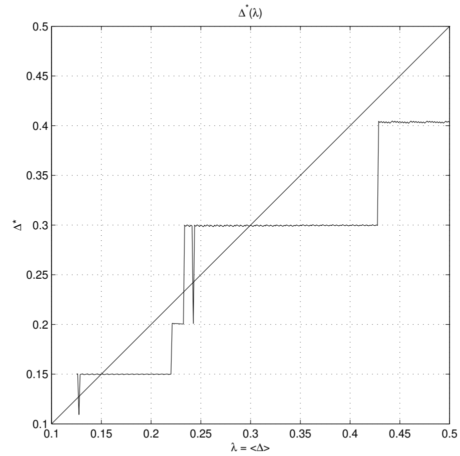

Figure 11: Dependence upon the Prior

scale. The method then searches over possible tempi and chooses the one which minimizes Eq. 16.

A resulting quantization ($\lambda = 0.4$) is given in Figure 12. The output differs from the desired notation in the 3. segment, which is quantized as (323) instead of a triplet (111). This is to be expected since a simple grid quantization schema can not recognize different notations which require different subdivisions. The vector quantization does not suffer from this drawback and produces exactly the desired notation (Fig. 13).

We have also tested the algorithm on data recorded from a solo piano performance (Fig. 14). The melody is actually an accompaniment melody which has only a regular beat (3 eight notes in each segment) however it is very expressively played and is difficult to quantize for a simple quantizer. This example compares the vector quantizer ($\gamma = 1.01$) with an onset and ioi quantizer which have the same degree of freedom (i.e. uses the same code-

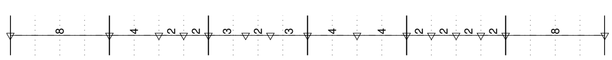

Figure 12: Quantized Rhythm (Grid quant.)

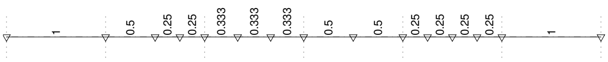

Figure 13: Quantized Rhythm (Vector quant.)

vectors).

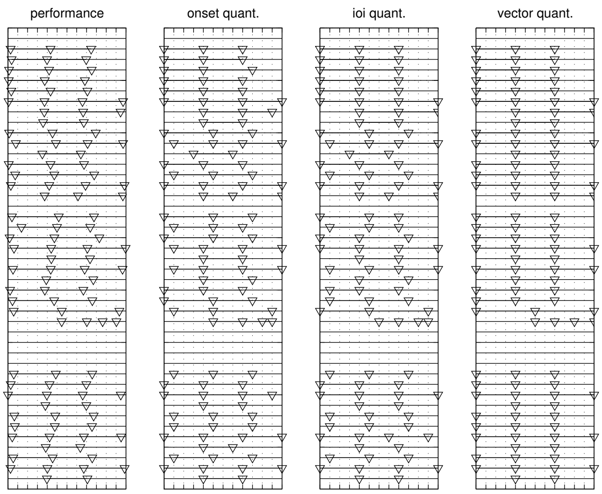

Figure 14: Quantization of real performance

The original performance suggests that the onsets are indeed correlated (we observe shifts in the same direction). By controlling the complexity with the prior and coding the correlation structure in the likelihood, the regularized vector quantizer produces a quantization which is very close to the original score.

6 Discussion and Conclusions

In this article, we developed a vector quantization schema for quantization of musical rhythms. We made use of Bayesian statistics where we proposed prior and expressive noise distributions based on observations from an perceptual experiment. It must be noted that the form of the likelihood and prior proposed in this paper resulted from subjective observations which are to be validated experimentally. Nevertheless, an advantage of this approach is that since the assumptions are stated as probability distributions, the parameters can be learned, if more experimental data is available.

In the simulations, we have observed that the complexity prior plays an important role when the subdivision schema is long (e.g. there are many small subdivisions). Then the particular choice of the correlation matrix does not play a crutial role. This is to be expected, since the volume of the space, where the quantization of an onset quantizer differs from an ioi quantizer goes to zero in lack of a complexity prior when there are increasingly many codevectors.

An important note is that in the derivation we did not used any other attributes of notes (e.g. duration, pitch), which can give additional information for quantization. This information can also be integrated into the framework by modifying the likelihood and prior suitably. Current research is directed towards investigation of this issue.

Acknowledgements

This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Dutch Ministery of Economic Affairs.

References

Carlos Agon, Gérard Assayag, Joshua Fineberg, and Camilo Rueda. Kant: A critique of pure quantification. In *Proceedings of the International Computer Music Conference*, pages 52–9, Aarhus, Denmark, 1994. International Computer Music Association.

E. F. Clarke. Structure and expression in rhythmic performance. In P. Howell, I. Cross, and R. West, editors, *Musical structure and cognition*. Academic Press, Inc., London, 1985.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, 1991.

P. Desain and H. Honing. Quantization of musical time: a connectionist approach. In P. M. Todd and D. G. Loy, editors, *Music and Connectionism.*, pages 150–167. MIT Press., Cambridge, Mass, 1991.

P. Desain, H. Honing, and K. de Rijk. The quantization of musical time: a connectionist approach. In *Music, Mind and Machine: Studies in Computer Music, Music Cognition and Artificial Intelligence*, pages 59–78. Thesis Publishers, Amsterdam, 1992.

Peter Desain, Rinus Aart, Ali Taylan Cemgil, Bert Kappen, Huub van Thienen, and Paul Trilsbeek. Robust time-quantization for music. In *Proceedings of the AES 106th Convention*, page (in submission), Munich, Germany, May 1999. AES.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., New York, 1973.

H. Christopher Longuet-Higgins. *Mental Processes: Studies in Cognitive Science*. 1987. 424p.

J. Pressing and P. Lawrence. Transcribe: A comprehensive autotranscription program. In *Proceedings of the International Computer Music Conference*, pages 343–345, Tokyo, 1993. Computer Music Association.