Measuring the Similarity of Rhythmic Patterns

Jouni Paulus
Tampere University of Technology
Institute of Signal Processing
P.O. Box 553, FIN-33101 Tampere, FInland
+358 3 3115 4790
paulus@cs.tut.fi

Anssi Klapuri
Tampere University of Technology
Institute of Signal Processing
P.O. Box 553, FIN-33101 Tampere, FInland
+358 3 3115 2124
klap@cs.tut.fi

ABSTRACT

A system is described which measures the similarity of two arbitrary rhythmic patterns. The patterns are represented as acoustic signals, and are not assumed to have been performed with similar sound sets. Two novel methods are presented that constitute the algorithmic core of the system. First, a probabilistic musical meter estimation process is described, which segments a continuous musical signal into patterns. As a side-product, the method outputs tatum, tactus (beat), and measure lengths. A subsequent process performs the actual similarity measurements. Acoustic features are extracted which model the fluctuation of loudness and brightness within the pattern, and dynamic time warping is then applied to align the patterns to be compared. In simulations, the system behaved consistently by assigning high similarity measures to similar musical rhythms, even when performed using different sound sets.

1. INTRODUCTION

> *Music is composed, to an important degree, of patterns that are repeated and transformed. Patterns occur in all of music's constituent elements, including melody, rhythm, harmony, and texture.* —Rowe, [1, p.168]

Pattern induction and matching plays an important role in music analysis and retrieval. Especially melodic fragment matching has received much attention in recent years [1,2,3,4]. However, measuring the similarity of *rhythmic* patterns has been almost a neglected problem. Work on the computation analysis of musical rhythms has concentrated almost entirely on beat detection and time quantization (see [5,6] for recent examples). Measuring the similarity of rhythmical patterns can be applied e.g. in musical database searches and in music context analysis in general [7].

It is intriguing to ask what makes two rhythms similar or dissimilar from a perceptual point of view. The problem has been addressed by musicologists in experiments, where rhythmic patterns were presented for human listeners for similarity judgments or for reproduction [8,9]. Obtained dimensions of dissimilarity have been interpreted to be e.g. "meter", "rapidity", "uniformity–variation", "simplicity–complexity" etc. [8]. A problem with these findings is that it is very difficult to encode and quantify them into a computer model. In following, a more pragmatic approach is taken.

The aim of this paper is to propose a method for measuring the similarity of two rhythmic patterns which are performed using arbitrary drum/percussive sounds, and presented as two continuous acoustic signals. A preliminary process estimates the musical meter and flags pattern boundaries. This is followed by the actual similarity measurements. No *a priori* knowledge of the rhythmic pattern classes is involved in the comparison. Thus the method is not confined to e.g. Western music.

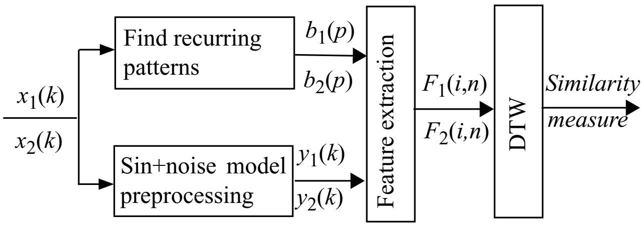

Figure 1. Overview of the system.

The task described above can be decomposed into a number of smaller requirements. First, two identical rhythmic patterns have to be recognized as similar even when played with different sounds. This has to do with the acoustic features used to describe the signal. Secondly, the patterns have to be aligned in time and tempo differences have to be reconciled. The common approach using hidden Markov models (HMMs) is not appropriate here, since only one instance of both rhythms is given, i.e., we are not aiming to recognize predetermined rhythm classes, but to compare two individual data sets. Also, the duration model of conventional HMMs is very loose, basically allowing only exponentially decaying distributions. For these reasons, dynamic time warping (DTW) was employed. DTW allows a certain amount of flexibility in time alignment, and it has been successfully used to handle a third subproblem of rhythmic pattern matching: musical variations. Similarity measurements must be robust to inserting, deleting, and substituting reasonable amounts of atomic elements.

Dynamic time warping is a dynamic programming algorithm that is based on sequental decision process. It has been originally used in template matching in speech and image pattern recognition since 1960's. Later on it has been replaced by HMMs in speech recognition. Dynamic programming has been successfully used in matching melodic patterns. Dannenberg used it for real-time accompaniment of keyboard performances [2]. The approach was further developed by Stammen and Pennycook [3]. More recent systems have sought mechanisms for pattern induction from repeated exposure, followed by pattern matching [1].

Overview of the system to be presented is shown in Figure 1. The different modules, preprocessing, pattern segmenting, feature extraction, and the DTW are now separately discussed.

2. METHODS

2.1 Pre-processing

An optional preprocessing step in the system is preprocessing with a sinusoidal model [11]. When analyzing percussive rhythms in real-world musical signals, it is advantageous to suppress the other (pitched) musical instruments prior to rhythm processing. Drum sounds in Western music typically have a clear stochastic noise component [10]. In addition, some drums have strong harmonic vibration modes and they have to be tuned. In the case of tom toms, for example, approximately half of the spectral energy is harmonic. Nevertheless, these sounds are still recognizable based on the stochastic component only.

A sinusoids plus noise spectrum model was used to extract the stochastic parts of acoustic musical signals. The model, described in [12], estimates the harmonic parts of the signal and subtracts them in time domain to obtain a noise residual. Even though some non-drum parts of signal end up to the noise residuals $y_1(k)$ and $y_2(k)$, the level of drums in relation to other instruments is considerably enhanced. The amount of non-drum sounds in the residual does not complicate the distance measuring too much since we are not interested in individual events, but in the entire rhythmic sensation.

2.2 Pattern Segmenting

An essential step before similarity measurements is to segment the continuous time domain signal into chunks that represent patterns. A brute force matching of all possible patterns of all lengths would be computationally too demanding.

Pattern segmenting is a part of a rather complicated musical meter estimation process, which is more or less independent of the subsequent similarity measurements. Earlier algorithms for automatic meter extraction have been developed e.g. by Brown and Temperley [13,14]. The estimator proposed here has not been previously published and is therefore now briefly introduced. The module takes the acoustic musical signal without preprocessing as input, and outputs the lengths of the tactus (beat) and the musical measure. The latter is interpreted as the rhythmic pattern length. Also, pattern phase is estimated, in order to be able to list a vector of pattern boundary candidates $b_1(p)$ and $b_2(p)$.

2.2.1 Mid-level Representation

A signal model is used which retains the metric percept of most musical signals while significantly reducing the amount of parameters needed to describe the signal. Only amplitude envelopes of the signal at eight sub-bands are stored. The general idea that rhythmic percept is preserved with this signal model has been earlier motivated by Scheirer in [15].

First, a bank of sixth-order Butterworth filters is applied to divide the input signal into eight non-overlapping bands. The lowest band is obtained by lowpass filtering at 100 Hz cutoff, and the seven higher bands are distributed uniformly on a logarithmic frequency scale between 100 Hz and half the sampling rate. Magnitude responses of the filters sum approximately to unity, and group delays of the filters are compensated for.

At each subband, the signal is half-wave rectified, squared, and decimated by factor 45 to 980 Hz sampling rate. Then a fourth-order Butterworth lowpass filter with 20 Hz cutoff frequency is applied to obtain the amplitude envelope of the signal at each frequency channel. Finally, dynamic compression is applied to obtain compressed amplitude envelopes $v_c(k)$ at channels c at time k:

$$v_c(k) = \ln[1 + Jz_c(k)], \qquad (1)$$

where $z_c(k)$ is the signal before compression and J=1000 is a constant. The value of J is not critical, but merely determines the dynamic range after compression and ensures that numerical problems do not arise. The amplitude of the original wideband input signal $x(k)$ is controlled by normalizing it to have zero mean and unity standard deviation before any of the described processing takes place.

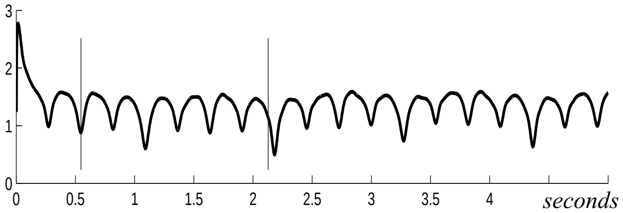

Figure 2. A typical instance of the function s(τ). The actual tactus and measure periods are indicated with vertical lines.

2.2.2 Periodicity Detection

Envelope signals $v_c(k)$ at each frequency channel are subject to periodicity analysis. For this purpose, we employ an algorithm which has been originally proposed by de Cheveigné and Kawahara for fundamental frequency estimation [16]. First, a difference function is formed:

$$d_c'(\tau) = \sum_{k=1}^{K} [v_c(k) - v_c(k+\tau)]^2 \quad , \qquad (2)$$

where K=4900 is the size of the time frame. In the decimated sampling rate, this corresponds to five seconds. The function is then mean-normalized to obtain $d_c(\tau)$:

$$d_c(\tau) = 1 \qquad \text{for } \tau=0 \qquad (3)$$

$$d_c(\tau) = \frac{d_c'(\tau)}{\frac{1}{\tau}\sum_{i=1}^{\tau} d_c'(i)} \qquad \text{otherwise}$$

This function is closely related to the inverse of the autocorrelation function, but was found to behave much more nicely due to the normalization. Minima of $d_c(\tau)$ indicate periods.

The bandwise functions $d_c(\tau)$ are then summarized over channels

$$s(\tau) = \sum_{c=1}^{8} A_c d_c(\tau) \qquad (4)$$

where A_c is the inverse of the minimum value of $d_c(\tau)$ over τ at channel c. The value A_c correlates strongly with the strength of the periodicity at channel c, and brings an important performance improvement by implementing an adaptive weighting of different frequency channels.

The function $s(\tau)$ serves as the source of information for musical meter estimation. Figure 2 illustrates a typical instance of $s(\tau)$ for a piece from *soft rock* genre (BeeGees: *Alone*). The actual beat and pattern periods are indicated with vertical lines. Please note that dips in $s(\tau)$ indicate periods.

2.2.3 Selecting Tatum, Tactus, and Measure Lengths

Musical meter is estimated at three levels: tatum, tactus (beat), and the musical measure. The term *tatum*, or, time quantum, refers to the shortest durational values in a musical composition that are still more than incidentally encountered. The other durational values (with few exceptions) are integer multiples of the tatum. *Tactus* is perceptually the most prominent metrical level, also known as *beat*, or the foot tapping rate. Musical measure is a still higher metrical level, correlated with the harmonic change rate, and most importantly, can be used to define the rhythmic pattern length. All the three levels are required to find the musical measure boundaries, i.e., the patterns.

Selection of the tatum period is done in a straightforward manner. We denote by $S(f)$ the discrete Fourier transform of $s(\tau)$. Tatum is determined according to the maximum of the function $\sqrt{f} \times S(f)$ in the range between 1.7 Hz and 20 Hz. Tatum period is the inverse of the frequency corresponding to the maximum value. The rationale behind weighting with $\sqrt{f}$ is to implement a proper preference towards higher frequencies. Otherwise e.g. the tactus or measure period may be detected.

Tactus period is calculated in a probabilistic manner using three probability distributions. The main likelihood function is obtained from $s(\tau)$, it is, from the observation. The probability of a tactus period τ given $s(\tau)$ is defined to be proportional to:

$$P[\tau|s(\tau)] \propto \frac{1}{s(\tau)}\,. \tag{5}$$

The concept of proportionality has to be used, since the integral over $1/s(\tau)$ is not guaranteed to sum to unity. The above likelihood is then multiplied by *a priori* probabilities measured from actual data by several authors. As suggested by Parncutt [17], we apply log-normal distribution for tactus periods, written

$$P_0(\tau) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}\left[\log_{10}\left(\frac{\tau}{\mu}\right)\right]^2\right\} \tag{6}$$

where the average tactus period μ was set to 600 ms and σ=0.25. The third probability distribution governing beat period probabilities comes from the tatum information. The conditional probability distribution for tactus periods τ given tatum period τ_0 is defined as a mixture of Gaussian distributions, written

$$P(\tau|\tau_0) = \sum_{m=1}^{9}\frac{a_m}{\sqrt{2\pi}\sigma_0}\exp\left\{-\frac{1}{2}\left[\frac{\tau - m\tau_0}{\sigma_0}\right]^2\right\} \tag{7}$$

where σ_0=0.3 and the weights for different multiples m of tatum,

$$a_m = \frac{1}{25}[4, 4, 3, 4, 1, 3, 1, 3, 2]\,. \tag{8}$$

The exact weight values are not critical, but simply realize a tendency towards binary or ternary subdivisions of the tactus. For example, the tatum multiples m={1,2,4} have a weight 4/25, but m=5 is assigned a weight 1/25 only. The overall likelihood of different tactus periods τ is then

$$P(\tau) \propto P[\tau|s(\tau)]P_0(\tau)P(\tau|\tau_0)\,, \tag{9}$$

the maximum of which indicates the most likely tactus period.

Finally, the musical measure length which indicates the patterns, is calculated in a manner analogous to tactus estimation. Likelihood for different measure lengths τ given $s(\tau)$ is obtained directly from Eq. (5). *A priori* probability distribution for measure lengths is calculated from Eq. (6) by substituting τ=2.2 and σ=0.4. Finally, the conditional probability of different measure lengths τ given the estimated tactus period is calculated using Eq. (7), where tactus period is substituted for τ_0, and $P(\tau|\ \tau_0)$ gives the conditional probability. The three probabilities are combined according to Eq. (9).

For pattern extraction, we need, not only the pattern length, but also its phase. This turned out to be even more difficult than finding the pattern period. A simple yet satisfactory solution was to construct a signal, where impulses are placed at pattern length distance apart. This signal is then correlated with the compressed amplitude envelope $v_c(k)$ at the lowest frequency channel, c=1. The highest maximum in the resulting correlation function was used as a temporal anchor for pattern beginning points.

2.3 Acoustic Features for Similarity Judgments

As shown in Fig. 1, the actual similarity measurement module gets as input the two noise residual signals $y_1(k)$ and $y_2(k)$, and a list of pattern boundaries. Because the pattern segmenting stage is not guaranteed to be 100 % reliable, a couple of most probable pattern lengths and phases are considered, one at a time, and two most similar patterns are used to determine the similarity measure. However, in Simulations section the pattern segmentation and similarity measurement stages are separately evaluated.

After we have isolated one pattern from both signals, acoustic feature extraction takes place in a series of consecutive 23 ms time frames. There was no significant performance difference between 23 ms and 46 ms frame lengths, though. The frames are Hanning-windowed and adjacent frames overlap 50 %.

2.3.1 Calculation of Features

The two most fundamental perceptual features of a individual rhythmic events (in addition to their timing) are their perceived loudness and brightness. Most musical rhythms, if not all, can be identifiably played using only these two dimensions. In addition to these two, an attempt was made to utilize the timbre information.

Loudness was modeled by calculating the mean square energy of the signal in one frame, and then by taking a natural logarithm to better correspond to the perceived loudness. More exactly,

$$L = \ln\left\{1 + \frac{J}{K}\sum_{k=1}^{K}[y(k)]^2\right\}, \tag{10}$$

where K is the frame size and the value J=1000 is used for the same practical purpose as in Eq. (1).

Spectral centroid has been found to corresponds to perceived brightness of sounds. It is defined as the balancing point of the spectral power distribution, and is typically calculated as the first moment of the magnitude spectrum. As suggested by Eronen in [18], a more robust feature is obtained by using a logarithmic frequency scale. Spectral centroid is here calculated as follows. First, the short-time power spectrum of the signal is calculated. Then a vector $U(b)$ is formed which contains the energies at sixth-octave frequency bands b, however, limiting the minimum bandwidth to one spectral line at the low frequencies. Spectral centroid is then

$$SC = \frac{\sum_{b=1}^{B}[f_c(b)U(b)]}{\sum_{b=1}^{B}U(b)}, \tag{11}$$

where $f_c(b)$ is the center frequency of band b in Hertz, and B is the number of bands in $U(b)$. Finally, centroid values at the linear frequency scale (Hz) are warped to a logarithmic scale simply by using $\ln(SC)$ as a feature.

Mel-frequency cepstral coefficients (MFCC) were extracted by applying the discrete cosine transform to the log-energy outputs of a mel-scaling filterbank [20]. The filterbank was implemented by calculating a discrete Fourier transform for the windowed waveform, and then simulating 40 triangular bandpass filters having equal bandwidth on the mel-frequency scale [19]. The zeroth cepstral coefficient is discarded, and a the next 15 coefficients are catenated to the feature vectors.

2.3.2 Normalization of Feature Vectors

The above calculated features reflect the absolute “tone color” and loudness in each individual time frame. As such, the features are not appropriate for rhythmic similarity measurement without normalization. Rhythmic events are perceived in their context and in relation to each other. A sound may take the role of a “bass drum” just because it is lower than the neighbouring events, not because of its absolute timbre. This allows musicians to reproduce rhythms with highly varying means, e.g. by tapping, scat-singing, or by playing with drums. In following, we propose normalizations which transform the absolute feature values to a relative representation.

The energy feature $L(n)$ over frames n=1,...,N is normalized by subtracting the minimum value of $L(n)$ over time, and scaling the resulting curve to have a unity variance. This can be written

$$L'(n) = [L(n) - L_0] / \sigma_L \tag{12}$$

where L_0 is the minimum value of $L(n)$ over n and σ_L is the standard deviation of $L(n)$.

The normalized loudness feature $L'(n)$ is used to weight the other features, in order that the tone color of soft or quiet segments would not influence on the similarity judgments too much. The weight vector $\lambda(n)$ is equal to $L'(n)$ divided by the sum over $L'(n)$.

The normalization of the other features, spectral centroid and mel-cepstrum coefficients, is performed as

$$SC'(n) = \frac{\lambda(n)}{\sigma_{SC'}}\left\{SC(n) - \sum_{m=1}^{N}[\lambda(m)SC(m)]\right\}, \tag{13}$$

where $\sigma_{SC'}$ is the standard deviation of $\lambda(n)SC(n)$. The resulting normalized features have zero mean and unity variance over time, and have been weighted with $\lambda(n)$.

Each individual feature over time has now been normalized to similar mean and range, and can be later weighted in a controlled manner in relation to other features. Also, the absolute tone color is discarded, modeling only deviations up/down from the average value. In this way, a tapped rhythm produces a feature vector which is similar to that produced by playing drums.

The normalized feature vectors are collected to a matrix $F(i,n)$ which contains feature vectors over the time range of the pattern,

$$F(i, n) = \begin{bmatrix} L'_1 & SC'_1 & \boldsymbol{MFCC'_1} \\ \ldots & \ldots & \ldots \\ L'_n & SC'_n & \boldsymbol{MFCC'_n} \end{bmatrix}, \tag{14}$$

where i is the index of each feature, and n is the frame index.

2.4 Dynamic Time Warping

Two feature vector sets $F_1(i,n)$ and $F_2(i,n)$ are matched using dynamic time warping (DTW). The DTW matches the two data sets by trying to find an optimal path through a matrix of points representing all the possible time alignments between the feature vector sets. The *template* feature vectors represent the row coordinates and the *unknown* feature vectors represent the column coordinates in the matrix. An example of a such matrix and the optimal time alignment path is illustrated in Figure 3.

2.4.1 Local Path Constraints

The main idea of DTW is that it allows some amount of adaptability when matching discrete data points of the two patterns. The adaptability is achieved by allowing the path to vary the rate at which it goes through the two patterns. The non-warping method would always match feature vectors at the same indices from both patterns. DTW for its behalf, allows the patterns to differ in length and still it finds the best fit for them. Some kind of path constraint is necessary, because it is not a good idea to allow the path to proceed randomly to a next point. Usually the set of possible next points is limited by local path constraints so that the local time warps become smaller.

The three different local path constraint types that were tried in this implementation are shown in Figure 4. They were chosen from among the eight different types presented in [19]. Type 1 local path constraint allows such transitions that path proceeds precisely one time frame only in template features or only one time frame in unknown features or one in both at the same time. This constraint is the most loose of all, because it allows the path even e.g. to proceed first horizontally through the matrix and then continue to the end point with only vertical steps. Considering what this means in terms of time warping, the path is quite inappropriate for template matching, because it is very improbable situation that the first feature vector in template matches almost all of the unknown pattern and all the rest of the template feature vectors are matched with just the last one of the unknown pattern.

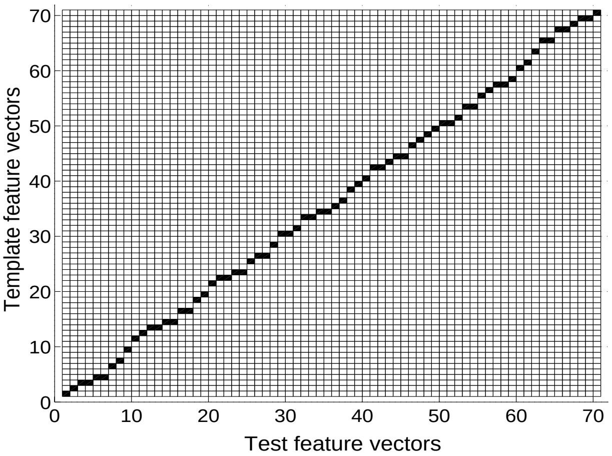

Figure 3. An example of DTW matrix and an optimal path.

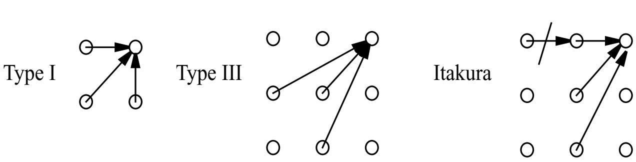

Figure 4. Local path constraints.

Type 3 allows only transitions which proceed in time frames both in template feature vectors and in unknown feature vectors. This type allows the matching to skip one feature vector either in template or in unknown, but not in both at the same time.

The third local path constraint type used here is called Itakura, first presented in [21]. Every transition proceeds in time frames in unknown feature vectors. Transition does not have to proceed in time frames in template feature vectors, but two consecutive horizontal steps are forbidden. This way it is possible to avoid situation where path gets stuck on a horizontal direction.

With type 3 and Itakura local path constraints it is implied to use also global constraints to limit the area which we must go through while searching for the optimal path. This is because the minimum path slope is 0.5 and the maximum slope is 2. Taking this fact and fixed start and end point of path in to consideration, the allowable region on which the path can reside is a parallelogram. The aim of the area reduction is simply to reduce the amount of possible paths and therefore the amount of needed calculations. It is possible to use similar global limitation with type 1 local path constraint, but it is not directly implied by the local path constraint.

2.4.2 Path Length

The total length of the optimal time-warped path to a certain point in the matrix is defined recursively by Eq. (15) below. The local path constraint used in this equation is type 3. The formulas for the other paths are analogous and can be found in [19]. The length of the optimal path to point *C*(*n,m*) is defined so that it has the smallest cumulative sum consisting of the feature vector difference cost *D*(*n,m*) and the minimum of sum consisting of the path length to the previous point and transition cost *T* from that point:

$$C(n, m) = D(n, m) + min\left\{ \begin{array}{l} C(n-1, m-1) + T(1, 1) \\ C(n-1, m-2) + T(1, 2) \\ C(n-2, m-1) + T(2, 1) \end{array} \right\} \quad (15)$$

$$D(n, m) = \sum_{i=1}^{I} \boldsymbol{W}(i)(\boldsymbol{F}_1(i, n) - \boldsymbol{F}_2(i, m))^2 \quad (16)$$

$$T(p, q) = \sqrt{p^2 + q^2} \quad (17)$$

In Eq. (16), vector *W* denotes the feature weight vector which controls how much a certain feature weighs in determing the similarity of two feature vectors and *I* is the number of different features. The transition cost *T* is here defined to be the Euclidean distance between the start and end point of the transition. The absolute magnitudes of the values in *W* determine the balance between the path cost *T* and feature vector difference cost *D*(*n,m*).

The final similarity measure between two rhythmic patterns is given by

$$S(\boldsymbol{F}_1, \boldsymbol{F}_2) = \frac{\sqrt{N^2 + M^2}}{C(N, M)}. \quad (18)$$

It is the theoretically shortest possible length divided by the cost of the optimal path. This makes it possible to compare the similarity measures of patterns of different lengths. With two identical patterns the similarity measure is 1, and the more the patterns differ the smaller the measure gets, gradually approaching to zero, but never actually reaching it.

2.4.3 The Core DTW Algorithm

The algorithm sequentially goes through the whole globally allowed area of the DTW matrix. For each point the optimal path length reaching it is stored as well as the link to the optimal previous point. This leads to the situation that every time the algorithm calculates the optimal path to a certain point, it already knows the path length to all possible preceding points. From these it then chooses the optimal one according to Eq. (15) and stores the resulting path length and preceding point information. This kind of steps are continued until the final point *(N,M)* is reached and the total path length is known. If it is not enough to know the length of the path, but also the actual path, it can be backtracked using the predecessor information stored in every point.

3. SIMULATIONS

3.1 Meter Estimation and Pattern Segmenting

Table 1 shows the statistics of the database used to evaluate the accuracy of the musical meter estimation and pattern segmenting algorithm presented in Sec. 2.2. Acoustic music signals were stored as single-channel, 44.1 kHz, 16-bit, pulse code modulated data. Tactus (beat) and musical measure (i.e. pattern) positions were manually annotated for one-minute long representative excerpts selected from each piece. The annotations were made by tapping along with the musical pieces, recording the tapping signal, and semiautomatically detecting the tapped time instants. Tactus and measures were separately annotated in different runs. Tactus could be more or less unambiguously judged for all the pieces. However, measure boundaries could be reliably marked by listening for a subset of the pieces only. Tatums were not annotated at all.

Table 1. Database for evaluating the meter estimation model.

Genre	Tactus annotated (# of songs)	Patterns annotated (# of songs)
Classical	85	–
Electronic/Dance	27	18
Hip Hop/Rap	12	8
Jazz/Blues	62	19
Rock/Pop	111	61
Soul/RnB/Funk	44	27
World/Folk	24	8
Total	**365**	**141**

In the simulations, the algorithm was given one 10-second excerpt from the beginning of each annotated one-minute period. The estimated tactus and measure periods were then compared to the manually annotated value. The estimated tactus and measure periods were defined to be correct, if the values deviated less than 10 % from the correct one.

The tactus periods given by the proposed algorithm were correct for 67 % of the 365 pieces. Most typical error was tactus period doubling. Estimated musical measure lengths (i.e., the pattern lengths) were correct for 77 % of the 141 pieces for which the measures was annotated, 17 % of the values were either half or double the pattern lengths, and 6 % were unclassified errors. However, it should be noted that the pieces for which the measure information could be annotated represent metrically more clear cases. This at least partly explains the performance difference between tactus and measure length estimation.

The estimated pattern phase was correct only in approximately half of the cases, suggesting a point of improvement in the system. In practice, this has more to do with computational efficiency, since reliable pattern comparison can be achieved by taking a couple of most prominent pattern length and phase candidates, performing the DTW for each candidate, and selecting the highest similarity value to the output.

3.2 Similarity Measurements

3.2.1 Similarity of Drum Patterns

A database of rhythmic patterns was used to validate the described similarity measurement approach. The database consisted of nine standard rhythm patterns with a couple of variations, totalling to 14 different patterns. The rhythms and the number of variations from each were: stomp, eight-note beat (x3), sixteenth-note beat (x2), triplet (x2), shuffle (x2), swing, waltz, samba, and songo. Variations were in the bass drum pattern for the 8th-beat, triplet, and shuffle rhythms, and in the hi-hat pattern for the 16th-beat rhythm.

Each of the 14 patterns was performed using three different sound sets, shown in Table 2. Swing rhythm makes an exception: it did not make sense musically to perform it with Set 3. The sounds were selected according to the principle that they would be as different as possible, yet exchangeable in different rhythmic roles. The differences between the bass drum sounds were small. However, the snare drum sounds are quite different, depending on whether played with stick or with brush, or at the rim of the drum. In the same manner, closed hi-hat, ride cymbal, and the shaker pro-

Table 2. Drum sets used in performing the rhythm patterns.

Drum set	Sounds involved		
1	bass drum	snare	hi-hat
2	bass drum	brush slap snare	ride cymbal
3	bass drum	cross stick	shaker

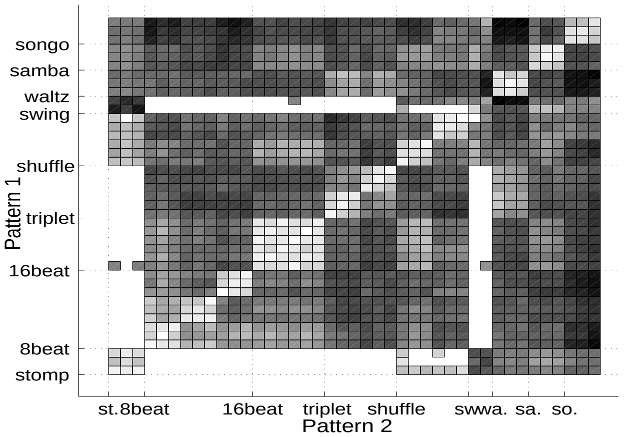

Figure 5. Calculated similarity measures for drum patterns.

duce rather different sounds, but are typically used in same rhythmic roles. An amateur musician performed the rhythms using Roland SPD-6 percussion pad together with two foot pedals.

The task of the system was to recognize the same rhythmic patterns as similar although performed using different sounds. Figure 5 shows the estimated similarities for each pair of the 41 performed patterns (3∗13+2∗swing). Each three consecutive samples represent identical rhythms, played with the three sets (except only two for swing). The whiter the area at the intersection of each two samples, the higher their estimated similarity. The white areas with a value missing are those for which the lengths of the two patterns differ by a factor greater than 2, in which case the local path constraint does not allow the comparison.

The illustrated similarity matrix was calculated using only the normalized spectral centroid as feature, and local path constraint 3. Preprocessing was not applied. In this experiment, similarity measurement was separately evaluated, taking the pattern boundaries from manually annotated time values.

The proposed system is successful in assigning a high similarity to same rhythms, despite of being performed with different sounds. Bass drum variations have the effect that the patterns practically appear as different rhythms. On the other hand, hi-hat variation in 16th-beat rhythm does not make a noticeable difference (a couple of hi-hat hits are omitted in the variation). This does not mean that the system would be purely bass-drum based, since among the 14 different patterns, six patterns have identical “bass drum/snare drum” patterns.

3.2.2 Performance of Different Features

Simulations were run to determine how well different features correlate with the rhythmic experience of a listener. In practice, the optimal weights $W(i)$ for different features in Eq. (16) were sought for. The weight values were determined by trying out different weight combinations and inspecting the resulting similarity matrix for the described rhythm database, and for complex musical signals, described in more detail in Sec. 3.2.3.

Normalized spectral centroid turned out to be clearly the best performing feature. After all, the most consistent similarity measures were produced by using this feature alone. However, it should be noted, that the normalized spectral centroid is actually an element-by-element product of the spectral centroid and loudness, as shown in Eq. (13). Loudness alone was somewhat successful, but using it together with the centroid only deteriorated the results.

Different numbers of MFCC coefficients were also evaluated as features. However, even after the normalization, MFCCs assigned high similarities to the patterns performed with identical sound sets, not to the patterns that were rhythmically similar.

As another observation, the path cost $T(p, q)$ in Eq. (15) had to be strongly weighted in relation to the acoustic features in order to discriminate between e.g. triplet and 16th-beat rhythms. With a high path cost weight, DTW allows the two patterns be of different lengths, and compensated for slight deviations in pattern beginning times, but punishes paths that are not straight lines through the matrix. In other words, steady time is constrained. Verification tests were performed which confirmed that the absolute lengths of the two patterns do not have a noticeable effect on the similarity judgment as long as the ratio of the lengths is between 0.5 and 2, required by the local path constraints. The best performing local path constraint in this experiment was the type 3.

3.2.3 Experiments with Complex Music Signals

In the last experiment, patterns taken from real-world musical signals were compared, using the database introduced in Sec. 3.1. Two patterns were taken from each of the annotated 141 songs using the manually annotated pattern boundaries. Then in-song and inter-song similarity measures were calculated, producing a matrix of 141x141 values, where the in-song measures are at the diagonal. The underlying assumption was that two patterns taken from a same song should be more similar than patterns from different songs, despite musical variations and the interference of other instruments.

The problem with this kind of evaluation is that it is difficult to know if the similarity is due to rhythmic characteristics. For example, MFCCs without normalization would model the absolute tonal color of the piece, bringing high in-song similarities but not necessarily because of the rhythm. For this reason, only the normalized spectral centroid was used as a feature, since we know that it does not retain any absolute features about the tonal color of a piece. Preprocessing with a sinusoidal model was applied.

Figure 6 shows the in-song distance for each of the 141 pieces, along with the average of inter-song distances calculated separately for each piece. The in-song similarity is consistently higher, but the difference is not large, most likely due to the other instruments and rhythmic variation.

4. SUMMARY AND CONCLUSIONS

The presented system was successful in extracting patterns from actual musical signals, and in assigning consistent similarity measures for drum patterns performed with different sound sets. The most successful acoustic feature for describing rhythmic patterns turned out to be the spectral centroid weighted with the log-energy of the signal. This vector was further normalized to have zero mean and unity variance over time. Dynamic time warping reconciled for tempo differences and slight beginning point deviations. However, a relatively high cost had to be assigned to the length of the time

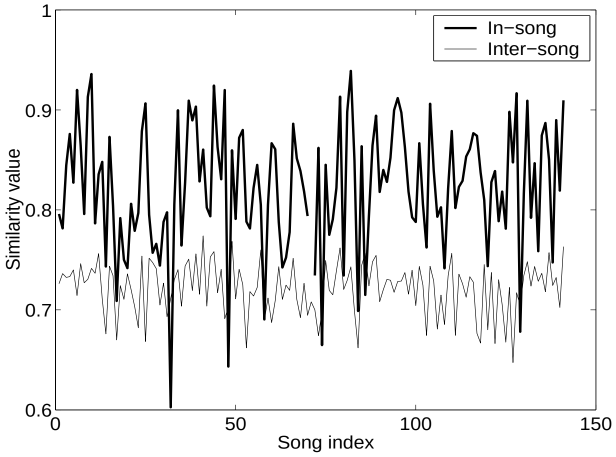

Figure 6. In-song and inter-song similarity measures for the database of 141 real-world musical pieces.

alignment path in order to constraint musical rhythms to steady time and to discriminate between binary and ternary rhythms.

5. ACKNOWLEDGMENTS

Matti Ryynänen performed the rhythmic patterns used in Sec. 3.2.1. The acoustic feature extractors were provided by Antti Eronen, Tampere University of Technology. The MFCC analysis was based on Malcolm Slaney's original implementation (http://rvl4.ecn.purdue.edu/~malcolm/interval/1998-010/). Sinusoidal modeling tools were provided by Tuomas Virtanen from Tampere University of Technology.

6. REFERENCES

[1] Rowe, R, "Machine Musicianship," MIT Press, Cambridge, Massachusetts, 2001.

[2] Dannenberg, R. B., "An On-Line Algorithm for Real-Time Accompanimen," In Proc. of the 1984 International Computer Music Conference, 193-198.

[3] Stammen, D.R., Pennycook, B. Real-time recognition of melodic fragments using the dynamic timewarp algorithm. ICMC Proceedings 1993, 232-235.

[4] Buteau, C. and Mazzola, G., "From contour similarity to motivic topilogies," Musicae Scientiae, Vol. 42, 2000.

[5] Dixon, S. E., "Automatic Extraction of Tempo and Beat from Expressive Performances," J. New Music Research, 30, 1, 2001, 39-58.

[6] Cemgil, A. T., Desain, P., Kappen, B. "Rhythm Quantization for Transcription," Computer Music Journal, Summer 2000, Vol 24:2.

[7] Chen, A. L. P, Chen, J. C. C., "Query by Rhythm, An Approach for Song Retrieval in Music Databases". In Proc IEEE Workshop on Continuous-Media Databases and Applications, 1998.

[8] Gabrielsson, A. Similarity ratings and dimension analyses of auditory rhythm patterns I. Scand. J. Psychol 14, 1973, 138-160.

[9] Powel, D.–J. and Essens, P., "Perception of Temporal Patterns," Music Perception, Summer 1985, Vol. 2, No. 4, 411-440.

[10] Fletcher, N. H. and Rossing, T. D., "The Physics of Musical Instruments," Springer–Verlag, New York, 1991.

[11] Serra, X., "Musical Sound Modeling with Sinusoids plus Noise," Roads, C. et al. (eds.) Musical Signal Processing, Swets & Zeitlinger Publishers.

[12] Virtanen, T., "Audio signal modeling with sinusoids plus noise," MSc thesis, Tampere University of Technology, 2000.

[13] Brown, J. C., "Determination of the meter of musical scores by autocorrelation". J. Acoust. Soc. Am. 94 (4), Oct. 1993.

[14] Temperley, D., "The Cognition of Basic Musical Structures". MIT Press, Cambridge, Massachusetts, 2001.

[15] Scheirer, E. D. "Tempo and beat analysis of acoustic musical signals," J. Acoust. Soc. Am. 103 (1), Jan. 1998, 588-601.

[16] de Cheveigne, A. and Kawahara, H. "YIN, a fundamental frequency estimator for speech and music," J. Acoust. Soc. Am. 111 (4), April 2002.

[17] Parncutt, R., "A Perceptual Model of Pulse Salience and Metrical Accent in Musical Rhythms," Music Perception, Summer 1994, Vol. 11, No. 4, 409-464.

[18] Eronen, A. "Comparison of features for musical instrument recognition". In Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2001.

[19] Rabiner, L., Juang B.H. Fundamentals of Speech Recognition. Prentice-Hall, New Jersey, 1993. 200-238.

[20] Davis, S.B., Mermelstein, P. "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," IEEE Transactions on Acoustics, Speech and Signal Processing, vol ASSP-28, No.4, 1980, 357-366.

[21] Itakura, F. Minimum Prediction Residual Principle Applied to Speech Recognition. IEEE Transactions on Acoustics, Speech and Signal Processing, vol ASSP-23, No.1, 1975, 67-72.