Exploration of techniques for automatic labeling of audio drum tracks' instruments

Fabien Gouyon, Perfecto Herrera
Music Technology Group, Pompeu Fabra University
{fabien.gouyon, perfecto.herrera}@iua.upf.es
http://www.iua.upf.es/mtg

Abstract

We report on the progress of current work regarding the automatic recognition of percussive instruments embedded in audio excerpts of performances on drum sets. Content-based transformation of audio drum tracks and loops requires the identification of the instruments that are played in the sound file. Some supervised and unsupervised techniques are examined in this paper, and classification results for a small number of classes are discussed. In order to cope with the issue of classifying percussive events embedded in continuous audio streams, we rely on a method based on an automatic adaptation of the analysis frame size to the smallest metrical pulse, called the "tick". The success rate with some of the explored techniques has been quite good (around 80%), but some enhancements are still needed in order to be able to accurately classify sounds in real applications' conditions.

1. Introduction

In this paper we define *drum tracks* as short audio excerpts of constant tempi containing few sets of percussive timbres: typically 5 to 10 seconds-long mixes of acoustic bass drums, snares drums, hi-hats, toms and cymbals. Our long-term objective is to design a set of analysis, transformation and browsing tools anchored in the musical contents of these signals, i.e. their rhythmic structures and their timbral features. Here we address the intermediary task of automatic description of drum tracks (see e.g. [15] and [3]). That is, given an excerpt, we aim at determining which percussive timbres are present, and at providing their timing occurrences; this, without making any assumption regarding the musical style.

When seeking a meaningful representation of audio signals, one must address the issues of characterization and segmentation (i.e. the "what is when?" issue). These concepts are tightly linked, and it is never clear which should come first. Indeed, segmenting temporally a signal into meaningful elements is obviously easier if we know what it's made of, and characterizing an event entails that it has boundaries. How could we meaningfully segment a signal before it has been categorized, or categorize it before it has been segmented? In this "chicken and egg" issue, focusing first on one task or the other implies explicit or implicit use of heuristics regarding the signal. In our case, we focus on *musical* signals of *constant tempi*, thus we assume that there always exists a relevant segmentation grid in regularly-spaced indexes that is inherent to the performance and corresponds to the smallest metrical pulse. These signals are solely made up of *percussive instruments*, therefore if the grid's gap and starting index are correctly determined, the resulting segments isolate coherent parts of percussive instruments signals around their indexes of occurrence in a scope determined by the smallest metrical pulse (typically 180ms), which is assumed to be sufficient for the purpose of characterizing these instruments.

The grid's gap is herein called the *tick* and corresponds to the smallest pulse implied by a performance on a set of drums (this concept is referred to as the "tatum" in [3], and the Yeston's "attack-point" concept (see [17]), as reported in [15] p.33). Occurrences of percussive events are supposed to match the tick grid with little approximation. We also assume that "new events" occur in the signal when *energy transients* are present. Unlike in the general polyphonic multitimbral case, it seems acceptable to state that occurrences of events in percussive music are linked to abrupt changes in the single energy parameter.

Therefore, we first focus on the segmentation of the signal, based on a prior detection of onsets; characterization comes in a second step. Handling these issues in the reverse way would be for instance to use metadata of the signal (e.g. 'this signal corresponds to the melody X played by the instrument Y'), and intend to adjust the segments boundaries to what we already know regarding what the signal is made of. An example of characterization before segmentation can be seen in e.g. [6] where a blind frame grid -which gap doesn't depend on the signal's content- is applied. Each frame is characterized in terms of chord type membership, and a subsequent segmentation in chords sequences is then applied. In that case, the heuristics lies in the knowledge of the signals to be processed: polyphonic audio signals with a strong harmonicity feature, i.e. audio streams in which it is relevant to assume that at each moment a chord is being played.

This paper uses a method to segment the signal respecting to a regular frame grid that is rhythmically relevant (see [7]). This segmentation in frames doing a good job in isolating significant part of percussive

events, we perform subsequent pattern matching algorithms over these frames to reach the objective of description. These algorithms are compared using an evaluation framework detailed hereafter.

1.1. Tick segmentation

In [7], we propose an algorithm to extract the smallest pulse implied in a performance on a set of drums. Let's remind the reader that the notion addressed here in not that of the beat –perceptively most prominent pulse–, rather a notion of pulse that most highly coincides with all onsets, at the metrical level of which corresponds the communication of important musical features (see [3]). In [7], we argue that peaks in the inter-onset intervals (IOIs) histograms of musical signals respect harmonic series, the gap of which we define as the tick. The proposed algorithm performs onsets detection, generates IOIs histograms and makes use of a powerful method of fundamental frequency extraction: the two-way mismatch algorithm proposed in [11].

An example of tick segmentation can be seen in Figure 1.

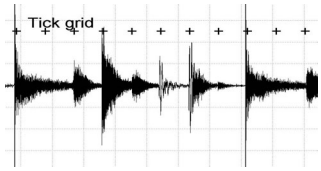

Figure1: Example of drum track tick segmentation grid

1.2. Percussive events characterization

Instruments sounds classification

There are two different approaches to automatic classification of musical instruments sounds. One of them is oriented towards perceptual classification (i.e. simulation of perceptual similarity judgments that can be obtained from human subjects in psychoacoustical experiments) whereas the other is oriented towards taxonomic classification (i.e. simulation of learned and culture-influenced judgments regarding the family, subfamily and type of instrument that can generate a test sound presented to a human or computer system). As we are here concerned with the second approach, the interested reader is addressed to [8] and [18] for more details on the perceptual approach and its associated techniques. Automatic taxonomic classification of sounds is an important functionality for the retrieval of audio files from large databases. In the case of song databases, location of specific instruments parts is an expected feature (though very difficult to be implemented given the current state of the art); in the case of sound samples databases, automatic labeling of samples after being incorporated in the database set is also a must (and this time it seems an achievable feat). The automatic labeling of sound events in audio drum tracks seems to be in a middle way, as it can include combinations of several sounds, but without the usual cluttering that is present in songs.

Even though there is an increasing literature on instrument classification (see [9] for a comprehensive review), very few works are concerned with the specific case of percussive instruments. In fact the most specific papers are aimed more to the classification based on perceptual similarity ([10], [14]). Although there are several techniques which provide a high percent of success when classifying isolated sounds (see e.g. [13] that addressed classification of entire isolated tones –several seconds long– of clean recordings), it is not clear that they can be applied directly and successfully to the more complex task of labeling monophonic phrases segments. With this aim, in [4], Brown addresses classification of short phrases (1-2 s) of recordings of varied sources. There is no segmentation of the signal in coherent events prior to the characterization, a frame grid with a fixed size (23 ms) is applied on the signal, each frame is treated independently of the others. The analyzed data is supposed to pertain to one and solely one of two classes ('oboe' or 'saxophone'), no switching from one instrument to another is envisaged. In [12], Marques and Moreno classify 0.2s-long segments. As for Brown, the assumption is implicit that all the frames constituting one excerpt pertain to solely one of the classes used for training. Moreover, none of these approaches intended to handle the issue of data that wouldn't pertain to any of the training classes (silences are removed from the tests sets).

Handling of sequential data

It is not clear yet how instruments classification methods like those commented above can be applied to audio mixtures without assuming a preliminary source separation, still not feasible. Nevertheless, it is of great interest to tackle the issue of automatically describing audio files that are more complex than monotimbral excerpts, which is the case of drum tracks (as several instrument sounds can overlap sometimes along the track, but in other times a single instrument is there). Indeed, a search (matching all the words) for "audio drum tracks download" objects on Internet with Google reports 27200 URLs of interest (October 19th 2001), which certainly corresponds to hundreds of thousands of drum tracks audio files "waiting to be analyzed".

In the task of classifying percussive sounds embedded in drum tracks, unavoidable steps are that of: (1) *segmenting* the drum tracks in "coherent" events, and (2) *choose a classification technique* to part these events in groups.

It is obvious that we can't assume that a segmentation technique will provide us with isolated clean drum occurrences similar to those of a training set. This has a great consequence if we choose a supervised classification technique. Indeed, even if

we assume that a database of isolated sounds could be representative of any drum track sounds, an issue would still lie in the comparison of (1) sounds that would have coherent boundaries regarding a *timbral* criterion (those of the database) with (2) sounds that would have coherent boundaries regarding a *rhythmic* attribute (the tick size) and can therefore differ from one drum track to another.

This has as consequence that we can't use descriptors that are relative to *entire* drum occurrences (e.g. "log attack time", "temporal centroid of the tone", or "spectral centroid of the tone", as in the percussive timbre space reported in [14]), then we have no choice but using descriptors of tick-length frames. Therefore, for classifying our specific data it seems that the following options are thinkable:

1) Set the frame size to the tick size and choose an unsupervised technique to detect grouping characteristics in the frames of a given excerpt. Here, no relation is supposed whatsoever amongst excerpts frames, nor between frames and the elements of a sound database.
2) Consider that the tick segmentation isolates the most characteristic parts of percussive instruments and apply supervised classification techniques: train a classifier with a database of labeled tick segments derived from audio drum tracks, classify incoming tick segments according to the resulting labeled regions in the projection space.
3) Divide all tick segments in smaller frames which size would be fixed (e.g. 15 ms) and independent of the excerpt's specific tick size. Train a classifier with labeled percussive instruments samples described at the frame level, the frame size being the same as that mentioned above. Here, the goal would be to characterize the evolution of features within a tick.

The eventual goal is to automatically associate a symbolic temporal sequence representing the occurrences of diverse percussive events to a given drum track. Choosing one or the other of the aforementioned options, this sequence will either represent the sole drum track *structure*, or its accurate *description*. That is, in the unsupervised framework, classes do not necessarily correspond to the same instruments in different drum tracks; what is achieved is the separation in clusters, not the assignation of meaningful labels. In this case, we propose to represent drum tracks by symbolic sequences (or "strings") constructed as the conjunction of several time-series of occurrences of different percussive timbres. As a trivial example, the string 'acdba' could stand for the sequence e.g. 'kick occurrence'-'nothing new'-'snare occurrence'-'hihat occurrence'-'kick occurrence'. The pattern-matching techniques we detail hereafter aim at providing symbolic sequences similar to that of Figure 2.

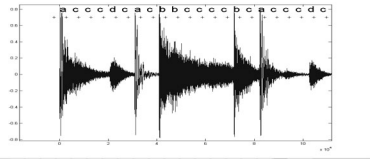

Figure 2: Drum track symbolic sequence

Here 'a' corresponds to an occurrence of kick, 'b' of a snare, 'd' of a hi-hats and 'c' to 'no new event'. The second 'b' is the sole artifact of the example.

On the other hand, in the supervised framework, it is intended from the very outset to recognize specific instruments, stating an universal membership of any family of percussive sounds to given regions in the representation space defined by the classification scheme's features.

In order to systematically evaluate the goodness of a given pattern-matching algorithm, we propose the following methodology.

2. Evaluation methodology

We developed an algorithm to generate random audio drum tracks, together with an exact score of these drum tracks. Given a tick value, at each position of the tick grid either a percussive instrument (with random amplitude) or silence is randomly assigned. Deviations from the exact tick grid positions are allowed and white noise is added to the signal for further (though in a quite crude way) approximating realistic performance data (see [7]). The resulting audio files constitute a useful audio material for automatic evaluation.

The evaluation process is the following:

- Audio drum tracks and scores are generated
- The segmentation in ticks is achieved over the drum tracks
- A tick computation is considered good if the computed tick is the same as the one given in the score of the track, with a possible deviation of ±1%.
- Features are extracted from tick frames (see below).
- The memberships of the frames to different classes is determined by either a supervised or unsupervised technique, and a symbolic sequence is generated
- These sequences are evaluated by comparison with the scores sequences.

When using unsupervised techniques, the assignation of a letter to a frame is arbitrary, for instance, an 'a' doesn't have to correspond to the same instrument in different drum tracks. Thus, in this case, the evaluation entails a brute-force string-matching algorithm: For each possible permutation of a computed sequence (if N is the number letters of the alphabet, there are N! possible permutations) the percentage of elements that aren't similar to those of the actual sequence is computed. Then, among the N!

percentages computed for a given sequence, the best one is chosen to be the percentage of matching of the computed sequence to the actual sequence.

To check the relevance of this process, we first evaluated the sole extraction of the tick: 1000 5-seconds drum tracks have been generated following the previous algorithm, four tick sizes being considered (250, 166, 124 and 83 ms). Comparing the ticks extracted from the audio signals to those given in the scores, the results are that 77.3% are correctly computed. An analysis achieved over real audio drum tracks has also been performed. The systematic evaluation is here more difficult as we don't have scores of the drum tracks that would provide the unambiguous knowledge of the tick. Here, the subjectivity of the listener enters obviously in the evaluation process, even more if the number of excerpts to evaluate is important. Nonetheless, it is interesting to mention the following results: Over 57 drum tracks, ranging from 2 to 10 seconds, made up of different bass drums, snares, hi-hats, cymbals, toms, corresponding mainly to reggae, funk, hip-hop and rock styles, comparing subjectively extracted minimal pulses with tick gaps and starting indexes yielded by the algorithm, the determination of the tick was considered good in 86% of the cases.

The automatic segmentation working sufficiently well, we now report on several pattern matching algorithms that have been run on the 1000 drum tracks mentioned above (summing 26680 occurrences of percussive sounds). The evaluation can be performed either on all the drum tracks or solely those segmented correctly.

3. Experiments

In a first step, we restrain the investigations solely to drum tracks made up of kicks, snares and hi-hats. The database of sounds used for the generation of drum tracks consists in 8 hi-hats, 9 kicks and 15 snares. It should be noted that we actually account for more than 32 sounds; indeed, the tick segmentation approach yields multiple variations of the same sound (estimation is around 5 variations per sound). In the generation algorithm, no simultaneous occurrences of instruments was generated, however, as the tick is generally shorter than the tones sizes, it should be noted that the issue of partially overlapping timbres still exists (e.g. a hit often occurs in the tail of another).

3.1. Features

Following the segmentation comes a process of recentering and windowing: the frame grid is shifted from half a tick, and each frame is multiplied by a Hanning window. There is no overlap of frames. The following descriptors are computed over each frame (they don't necessarily correspond to entire tones' descriptors):

- *Spectral kurtosis* (SK): This is a measure of how outlier-prone the spectrum is. Spectra which distribution are more outlier-prone than the normal distribution have kurtosis greater than 3; those that are less outlier-prone have kurtosis less than 3.

  $$sk = \left(\frac{1}{n\sigma^4} \sum (X - \overline{X})^4 \right) - 3$$

  where X is the magnitude spectrum of a frame, σ is the spectrum distribution standard deviation
- *Temporal centroid* (TC): This is the balance point of the absolute value of the temporal signal.
- *"Strong decay"* (SD): This feature is built from the non-linear combination of the frames energy and temporal centroid. A frame containing a temporal centroid near its left boundary and a strong energy is said to have a strong decay.

  $$sd = \sqrt{\left(1/tc\right) \cdot e}$$

  (where e is the energy of a frame and tc the temporal centroid)
- *Zero-crossing rate* (ZCR).
- *Spectral centroid* (SC): This is the balance point of the spectrum.
- *Energy* (E).
- *"Strong peak"* (SP): Intended to reveal whether the spectrum presents a very pronounced peak. The thinner and the higher the maximum of the spectrum is, the higher value takes this parameter.

  $$sp = \frac{\max(X)}{bW}$$

  (where X is the magnitude spectrum of a frame, bW is the bandwidth of the maximum peak in the spectrum above a threshold –half its amplitude)

3.2. Hierarchical clustering

Either being on-line or off-line,[1] a clustering process is based on proximity measures between elements and between groups of elements (see [16], p.358-378 for details on different proximity measures). Hierarchical clustering algorithms produce a hierarchy of clusterings that provides different degrees of agglomeration of the data, the number of clusters decreasing at each agglomeration step. The clustering produced at each step results from the previous one by merging two clusters into one, depending on the distance chosen. The process begins by assigning a cluster to each data vector, it ends when the whole data is agglomerated into a single cluster. The hierarchy of clustering can be explored in a dendrogram structure (see Figure 4).

Seeking an agglomeration in *four clusters*, and using deviations of the descriptors from their mean

[1] On-line procedures progressively adapt the clusters number, shapes and centroids according to incoming data, whereas off-line procedures process the data as a whole.

value, normalized by the standard deviation, we experimented the use of several descriptors, several elements-distances and several groups-distances (see [16]). A selection of the results are given in the table 1, where we illustrate the changing of features, then of elements-distance, then of groups-distance and then of features again.

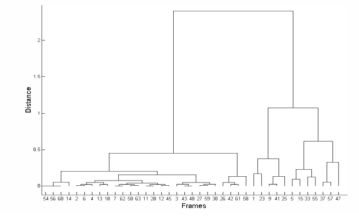

Figure 4: Hierarchical clustering dendrogram structure

3.3. Decision tree

We empirically noted that the parameter SD allows to separate frames of kicks and snares occurrences from the rest in the same manner: high values of SD correspond to occurrences of kicks and snares, and small values to the remaining frames (that is, occurrences of hi-hats and frames in which no new occurrence takes place). Therefore, we designed an algorithm that defines clusters in a step-by-step agglomerative manner. Decision steps are binary and taken respecting the last level of agglomeration of a hierarchical clustering scheme (elements-distances being Mahalanobis and groups-distances being Ward).

Frames are separated in class 1 and class 2 using the parameter SD. The class that shows the highest values of SDs is labeled Cl1, the other Cl2. Then we experimented different parameters to divide Cl1 data in class A and class B and Cl2 in class C and class D. A selection of the results are given in table 2.

3.4. K-means clustering

K-means clustering consists of first deciding the number of clusters that we need to get, and then running an algorithm that picks one "seed" case for each cluster. Then depending on its distances with the seeds, each case is reassigned a cluster membership and the seed case is updated as the centroid of its corresponding cluster. A function is computed that depends on the distances (e.g. Euclidean) between elements and centroids, and represents how well the centroids match the data. The process enters an iteration loop that terminates when a local minimum of the function is reached (see [5], p. 526). Though K-means does not perform an exhaustive search through every possible partitioning of the data, it provides results that overall perform quite well. Restraining tests to correctly segmented excerpts, we tested values of K ranging from 4 to 7 and the best results were obtained for K=4 (coincidentally with the number of a-priory categories of sounds: kick, snare, hihat, and no-instrument). From the table 3 it is clear that each cluster predominantly identifies one type of instrument (cluster 1 agglomerates the 84% of hihats, cluster 4 agglomerates 73% of kicks, and cluster 3 agglomerates 73% of no-instrument ("nothing")), but there is a problem with the snares, as cluster 2 only agglomerates a 54% of them whereas there is a 43% in cluster 4, apparently "confused" with kicks. The association between cluster and category is supported by the values of a Chi-square statistic (χ^2= 36303.134, d.f. 9, $p < 0.001$). The total matching between the algorithm output and the scores is 71.35%.

3.5. Fuzzy c-means clustering

The fuzzy c-means algorithm (like its "crisp" version, the c-means –or k-means) produces successive clusterings while trying to minimize a specific cost function. In the c-means algorithm, the data elements are assigned *exclusive* memberships with respect to a given number of clusters. Introduced in [19], the concept of fuzzy clusters stands in the assignation to each data element of a partial or *distributed* membership to each cluster. In the fuzzy c-means algorithm, these memberships are used as weights in the computation of the distances between data elements and centroids (see [2]).

Distributed memberships are assigned to the data elements until the process reaches convergence (i.e. improvement in the cost function smaller than a threshold), eventually the final memberships are exclusive (for the purpose of classification) and correspond to the highest memberships. For instance, if the memberships to 4 clusters of 7 frames are as in table 4, the resulting sequence is: 'b b a d a d b'. We tried several parameters and cluster numbers. A selection of the results are given in the table 5. One can see that the best clustering is performed with all the features and the correct number of clusters.

3.6. Linear Discriminant Analysis

A simple supervised technique is that of Linear Discriminant Analysis (LDA), an equivalent of multivariate analysis of variance for categorical variables. LDA attempts to minimize the ratio of within-class scatter to the between-class scatter and builds a definite decision region between the classes. It provides linear, quadratic or logistic functions of the variables that "best" separate cases into two or more predefined groups, but it is also useful for determining which are the most discriminative features and the most alike/different groups. Focusing on correctly segmented drum tracks, we checked that our data distributions were Gaussian and that their variances were similar, and then randomly selected a 65% of the sounds in order to derive a set of linear discriminant functions. They were then applied to the rest 35% in order to cross-validate the functions.

Results for this test set are shown in table 6. Though the overall success is 80%, there is a clear confusion phenomenon between snares and kicks that will require further study. Regarding the usefulness of variables, it seems that the most relevant are the temporal centroid, the spectral centroid, and the strong decay. The least relevant seem to be energy (which, on the other hand, is highly correlated with strong decay) and ZCR (which is highly correlated with spectral centroid), in fact the elimination of these two variables improved the snare discrimination, but at the cost of degrading a bit the classification of hihat and "no-instrument" categories.

3.7. Discussion

Clustering methods are sometimes considered as "second-league" methods that should be used cautiously (see [1]). However, reasons for exploring this approach can be found in the immediateness of results (against the time consuming parameter-setting "learning" phases that are required in most of the powerful supervised techniques). The other practical advantage is that a large proportion of realistic sounds extracted from drum tracks will be mixtures that cannot be easily labeled (and at the present moment we do not know whether a general "kick+hihat" category will be enough for that or we shall use different combinatory categories regarding some difficult-to-define criterion). In that respect, clustering techniques apparently seem more suitable than supervised learning categories. Anyway, it is still something to be studied as a next step of our research. On the other hand, a supervised technique will provide us with a direct labeling provided the adequate learning or estimation phase, and a faster assignment of labels than using clustering. If we take the performance of the LDA as the lowest estimation for a supervised technique, it seems that classification success is similar to that of clustering. Therefore, it is reasonable to expect an improvement when using more powerful techniques. Handling and labeling of multifarious mixtures can be efficiently done with Gaussian Mixture Models or Hidden Markov Models. Anyway, it should be noted that the percentages of supervised and unsupervised experiments should be compared cautiously. Indeed, it should be investigated what is the influence on both rationales of the representativeness of the instruments database used to generate tracks.

4. Future work

The present work illustrates the appropriateness of a set of audio descriptors for the task of percussive sound classification. We used a flexible framework suitable for the generation of drum tracks that allows systematic testing of classification techniques and features. Given the specific windowing method that we have used it is possible that some important features have been lost or corrupted (specifically, this could be the reason for the snare-kick confusion detected in some of the analysis). Improving the set of descriptors is therefore one of our next steps.

A couple of easy-to-do additional improvements will be the usage of more complex drum tracks, containing more diverse sounds (for example, toms or crash cymbals) and the creation of drum tracks that contain realistically overlapped sounds; indeed, it is frequent that e.g. hi-hats and kicks be played simultaneously. In that respect, we are currently designing an algorithm to generate drum tracks based on MIDI files data, that contain realistic mixtures of sounds, and where the labeling is given. This way we may get access to a suitable database as large and complex as specific experiments call for.

More realistic drum tracks will provide us with data actually pertaining with diverse degrees to several classes. It will be investigated whether fuzzy clustering permits to cope with this issue.

The supervised classification techniques that we have explored are not the most powerful, but have given an indication of the usability and limitations of our framework. Although we have obtained fairly good results, it is now clear that we need to jump towards techniques such as Gaussian Mixture models, Hidden Markov Models or Support Vector Machines if we want to improve the performance of our system.

5. Acknowledgment

The work reported in this paper has been partially funded by the IST European project CUIDADO.

6. References

[1] Aldenderfer, M., Blashfield and R., *Cluster Analysis*, SAGE Publications, Newbury Park, CA 1984.

[2] Bezdek J and Pal S.K., *Fuzzy models for pattern recognition*, IEEE Press, New York 1992.

[3] Bilmes J. *Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm.* MS Thesis/Dissertation MIT, 1993.

[4] Brown J., *Computer identification of musical instruments using pattern recognition with cepstral coefficients as features*. Journal of the Acoustical Society of America *105,* 1999.

[5] Duda R., Hart P. and Stork D., *Pattern classification*, John Wiley & Sons, New York 2001.

[6] Fujishima T. *Real-time chord recognition of musical sound: a system using Common Lisp Music.* In Proc. International Computer Music Conference, 1999.

[7] Gouyon F., Herrera P., and Cano P. *Pulse-dependent analyses of percussive music.* To appear.

[8] Grey J., *Multidimensional perceptual scaling of musical timbres*. Journal of the Acoustical Society of America *61*, 1977.

[9] Herrera P., Amatriain X., Batlle E., and Serra X. *Towards Instrument Segmentation for Music Content Description: a Critical Review of Instrument Classification Techniques.* In Proc. of International Symposium on Music Information Retrieval, 2000.

[10] Lakatos S., Beauchamp J., *Extended perceptual spaces for pitched and percussive timbres, J. Acoust. Soc. Am., 107 (5), 2000.*
[11] Maher J. and Beauchamp J., *Fundamental frequency estimation of musical signals using a two-way mismatch procedure*. J of the Acoust Soc of America *95,* 1993.
[12] Marques J. and Moreno P. *A study of musical instrument classification using gaussian mixture models and Support Vector Machine.* CRL 99/4 1999 Cambridge Research Lab.
[13] Martin K.D. and Kim Y. *Musical Instrument Identification: A pattern-recognition approach.* In Proc. of the 136th meeting of the Acoustical Society of America, 1998.
[14] Peeters G., Mc Adams S., and Herrera P. *Instrument sound description in the context of MPEG-7.* In Proc. of the International Computer Music Conference, 2000.
[15] Schloss A. *On the automatic transcription of percussive music - From acoustic signal to high-level analysis.* PhD Thesis/Dissertation CCRMA, Stanford University, 1985.
[16] Theodoridis S. and Koutroumbas K., *Pattern Recognition*, Academic Press, San Diego 1998.
[17] Yeston M., *The stratification of musical rhythm*, Yale University Press, New Haven 1976.
[18] Young F., *Multidimensional scaling: history, theory and applications*, Lawrence Erlbaum, 1987.
[19] Zadeh L., *Fuzzy sets*. Inform Control *8,* 1965.

Features		ZCR, E	E, SD	E, SD	E, SD	E, SD, SK
Elements distance		Mahalanobis	Mahalanobis	Euclidean	Mahalanobis	Mahalanobis
Groups distance		Shortest	Shortest	Shortest	Ward	Ward
Results	Excerpts with good ticks	59.2%	73.8%	71.6%	**81.1%**	79.4%
	All excerpts	53.7%	65%	63.5%	71.1%	70%

Table 1: Computed success rates of hierarchical clustering experiments

Features for dividing C11		SK	SK, ZCR	SK, SP	SK, SP
Features for dividing C12		E	E, TC	E, TC	E, SC
Results	Excerpts with good ticks	83.6%	78.6%	84.6%	**84.8%**
	All excerpts	73.2%	69.2%	73.8%	74.2%

Table 2: Computed success rates of decision tree experiments

	1	2	3	4	Total
hihat	**84.516%**	0.778%	14.679%	0.027%	100%
kick	3.528%	23.039%	0.208%	**73.225%**	100%
nothing	25.235%	0.514%	**73.737%**	0.514%	100%
snare	2.838%	**53.957%**	0.150%	43.054%	100%
Total	29.764%	21.368%	17.110%	31.758%	100%
N	7941	5701	4565	8473	26680

Table 3. Distribution percents of sounds into clusters by K-means clustering.

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7
Cluster 1	0.0013	0.0016	0.9842	0.0046	0.9963	0.0048	0.0005
Cluster 2	0.9959	0.9950	0.0037	0.0007	0.0008	0.0008	0.9984
Cluster 3	0.0020	0.0024	0.0015	0.0004	0.0003	0.0004	0.0007
Cluster 4	0.0008	0.0010	0.0106	0.9943	0.0026	0.9940	0.0003

Table 4: Example of memberships of some frames to 4 clusters in a fuzzy c-means clustering, after convergence.

Number of clusters		3	4	4
Features		SK, SP, SD, SC, E	SK, SP, SD, SC, E	SD, E
Results	Excerpts with good ticks	76.4%	**84.1%**	75.9%
	All excerpts	67.6%	73.6%	67.1%

Table 5: Computed success rates of fuzzy c-means experiments

	hihat	kick	Nothing	snare	%correct
hihat	2281	11	193	6	92%
kick	26	2031	1	407	82%
nothing	142	25	1165	3	87%
snare	23	823	0	1276	60%
All instruments					**80%**

Table 6: Confusion matrix and percent of correct classifications for a linear discriminant classificator