# Instantaneous Detection and Classification of Impact Sound: Turning Simple Objects into Powerful Musical Control Interfaces

**Nikolaos Stefanakis**
FORTH
Institute of Computer Science
70013 Heraklion, Crete, Greece
nstefana@ics.forth.gr

**Yannis Mastorakis**
FORTH
Institute of Computer Science
70013 Heraklion, Crete, Greece
University of Crete
Department of Computer Science
71409 Heraklion, Crete, Greece
jmastor@csd.uoc.gr

**Athanasios Mouchtaris**
FORTH
Institute of Computer Science
70013 Heraklion, Crete, Greece
University of Crete
Department of Computer Science
71409 Heraklion, Crete, Greece
mouchtar@ics.forth.gr

## ABSTRACT

This paper demonstrates an approach for achieving instantaneous detection and classification of impact sounds that the user produces while interacting with simple daily objects. Using a single microphone, the system is trained to recognize the differences in the resonant behavior of a plastic bucket, a box made of paper and an empty bottle of beer, as these objects are struck at different locations. The method employs a first-nearest neighbour classifier which is based on simple spectral features extracted from a very short segment of the acoustic signal. Tests performed illustrate that classification rates above 90% may be achieved with a system response around 5 ms or even less. While still perfectible, the presented work illustrates the potential in creating a generic system which would enable the users to turn costless objects into powerful music controllers and percussive instruments into Hyper-instruments, by training the system to respond to their disposable instruments and audio equipment.

## 1. INTRODUCTION

In the past two decades, the automatic detection and classification of percussive sounds from audio signals has attracted the interest from many different researchers. In most of the works, the primary motivation has been the demand to improve the efficiency of content-based management systems by looking into the rhythmic structure of musical pieces. While there is no standard method, there are numerous different approaches dealing with automatic transcription of drum sounds from monophonic or polyphonic music recordings, several of them exhibiting a very high performance. The work of Herrera et al. [1] provides a very good review of a set of classification techniques for isolated sounds and it was among the first attempts to propose a set of universal descriptors that are valid for a wide class of percussive instruments, as well as to find ways to visualize the relationship between these classes. In several studies that followed, the weight is put into the adaptation of the descriptor models to the content of the analysed audio file, exploiting therefore the repetitive nature of the drum patterns in these files [2, 3, 4, 5]. In the vast majority of these studies, an off-line application is considered, with an exception being that of Tanghe et al. [6], who considers a real-time streaming solution to drum detection.

In contrast to automatic transcription and audio queries, Human Computer Interaction systems (HCI) have real-time constraints and demand a fast system response. They also require reliable detection and identification of percussive sounds produced from the user, in an attempt to interact with him by providing some type of visual or acoustic feedback or by adapting some system parameters according to his performance. A first work considering real-time detectors of percussive music is that of Puckette et al. [7], later exploited by others in the context of an automatic accompaniment system [8] and a rhythmic tutoring system [9]. A system able to perform similar tasks was also recently presented by Şimşekli et al. [10], showing a good adaptability to different instruments and acoustic conditions. Finally, a beat tracking system which is based on real-time drum detection can be found in the work of Battenberg [11].

Recently, different products have been launched in the market, providing to the user the ability to control a sound synthesis process by interacting with simple daily objects. "Mooges" (http://mogees.co.uk), operates on the output signal of a contact microphone which is attached on the surface of the physical object. The system is able to track the user's gestures continuously and synchronously [12]. It provides immediate acoustic feedback with the intention to allow the user to learn how to interact with the physical object in order to improve his performance. "TableDrum" (by Dohi Entertainment), uses the built-in microphone of smart-phones and mobile devices. It operates based on a training stage where the user first records a few acoustic instances from different objects and then uses these objects in order to trigger a built in percussive synthesiser. The intention thus is to allow the system to learn how to respond to the user's gestures rather than the opposite. While this particular product is the closest example to the task that

we consider in this paper, we should state that we could not find any relevant bibliographic work and therefore, the methods that are exploited in this application are unknown to us.

In this paper, we use a single acoustic sensor (microphone) and we employ a low-cost onset detection and a nearest neighbor classification algorithm in order to simulate a real-time classification task. Similar to the case of "TableDrum", we employ a training phase for learning the variability within the different acoustic structures but we focus on the resonant behavior of a single acoustic object, as it is stroked at different *impact regions*. Results are provided for three simple objects of different material and size and the relation between classification accuracy and system latency is highlighted. The findings of this work are discussed under the perspective of applying the technique on real percussive instruments. This scenario is particularly interesting, as it would allow the possibility for the physical instrument to operate as a traditional percussion controller and more important, as it would enable the users to create an augmented sound to accompany the physical sound, turning thus their percussive instruments into Hyper-instruments.

## 2. CHALLENGE

When seeing detection and classification of percussive sounds from the perspective of a real-time application, one expects an obvious trade-off between latency and classification accuracy. The less is the latency that one would like to have, the less the amount of information that can be extracted from the acoustic event before assigning a label. Ideally, the time delay between the acoustic onset and the action produced by the computer should be imperceptible. This demand poses an important restriction to the length of the *analysis frame* which can be used for classification, from now on symbolized as $t_{fr}$. For obvious reasons, the system response can not be faster than $t_{fr}$.

How much can we then shorten the analysis frame and how much do we expect the classification performance to degrade? This depends on the nature of the acoustic instruments or objects that are used as well as on the complexity of the rhythmic pattern that is performed. For example, hand-claps [13] and finger-snaps [14] are optimal in terms of a fast system response because they last only for a few milliseconds. On the other hand, other objects will have significantly longer acoustic tails and this might degrade the process for an obvious reason; the tail of the previous strike will mask the onset of the new strike and the extracted acoustic features will be contaminated with "noise". While we can think of several approaches for resolving this ambiguity (e.g. source separation), it is shown in this paper that this problem may in a large degree be avoided by the choice of the physical objects that are involved in the process. Interestingly, simple costless objects of our daily lives seem to be very convenient for such tasks and moreover, their acoustic structures are optimal for achieving an instantaneous system response.

## 3. METHODOLOGY

### 3.1 Onset detection

As in many other approaches, our method for onset detection relies on measures of spectral energy on short audio segments which are called *frames*. We form frames by windowing the signal with a short-length Hanning window moving on a continuous time-grid with hop-size $h$. At each frame, the short-time Fourier transform (STFT) is calculated and the frequency bins with index $k$ corresponding to a specified spectral range $k \in [k_{min}, k_{max}]$ are used for further processing. A relatively good method for percussive sounds which exploits such features is the so called "percussiveness" measure, proposed by Tan et al. [15]. This method relies on the ratio of the magnitude of each frequency bin between the current frame and the previous frame. We have observed that the method responds well for a wide range of dynamic levels but the peaks of the detection function appear on a noise-floor which is prominent for causing false detections. In order to avoid such false detections, we employ an additional measure, $B$, which is equal to the L1 norm of the vector consisting of the magnitude of the frequency bins of the STFT in the previously defined spectral range, at the current frame. We accept candidate frame centres as onset locations only when conditions $A > A_{tr}$ and $B > B_{tr}$ are valid, where $A$ is the percussiveness measure and $A_{tr}$, $B_{tr}$ are empirically defined thresholds. To be noticed that measure $B$ is not only useful for onset detection; it may be also exploited as a measure of the intensity of the strike, which may in turn be used as an expressive parameter for controlling the synthesizer at the rear end of the process.

In order to facilitate onset detection further, we admit two basic assumptions; first, we assume that there is only one acoustic event happening at each time instant and second, that there is a minimum amount of time between two successive onsets, which we call the Minimum Anticipation Time and we symbolize it with $t_{ant}$. The parameter $t_{ant}$ may be used in order to disregard any detected onsets after a period of time less than $t_{ant}$ following the last detected onset. This is helpful in order to avoid a "double onset" due to ambiguities in the sound in the neighbourhood of a strong attack. While this may result in missed detections in the case of two rapidly played strokes, it is not a problem in this monophonic case.

### 3.2 Classification approach

Let $s[n] = s(nT)$ denote the discrete acoustic signal, sampled at a constant rate $Fs = 1/T$, which is input from the soundcard. The STFT of a percussive event which is detected at discrete time $jh$ may be written as

$$S_j[k] = \sum_{n=0}^{N-1} s[jh+n] e^{\frac{-2i\pi nk}{N}},\ k = 0, 1, ..., N-1 \quad (1)$$

where $j$ is the frame index, $h$ is the hop-size (used for onset detection) and $N$ is the length of the signal that is used for the STFT.

The raw data from the STFT of known acoustic events is used in order to construct a dictionary for each class, and

these dictionaries are to be used in order to classify unknown events. Each element of the dictionary is built by considering only a small subset of continuous frequency bins of index $k$ such that $f_{min} \leq \frac{kFs}{N} \leq f_{max}$, where $f_{min}$ and $f_{max}$ are minimum and maximum frequency limits which are the same for all classes. These frequency limits need not span the entire frequency range of the instrument; it is sufficient if they span all or some part of the frequency range of its most dominant acoustic modes. We may now denote the $K \times 1$ *input vector* associated to an onset detected at time $j$ as

$$\mathbf{x}_j = [S_j[k_{min}], ..., S[k_{max}]]^T, \quad (2)$$

where $k_{min}$ and $k_{max}$ are the smallest and largest index of the frequency bins that are taken into account. During the training process the input vector is normalized to have unity L2 norm and stored in the memory as the spectral *feature vector* representative of the $v$-th instance of the $i$-th class

$$\mathbf{a}_{i,v} = \frac{\mathbf{x}_j}{\|\mathbf{x}_j\|_2}. \quad (3)$$

where $\|\cdot\|_2$ denotes the L2 norm of a vector. We may now consider all $V_i$ different instances available from the $i$-th class in order to construct the class-specific *feature dictionary*

$$\mathbf{\Phi}_i^N = [\mathbf{a}_{i,1}, \mathbf{a}_{i,2}, ..., \mathbf{a}_{i,V_i}]^T, \quad (4)$$

where $N$ denotes the size of the STFT and therefore is representative of the length of each feature vector in the dictionary.

Observe that in the current process, the acoustic features are complex, including both magnitude and phase information. An alternative implementation would be to consider only the magnitude of the frequency response and to disregard the phase information. We will discriminate those two cases by referring to complex and real feature vectors and dictionaries respectively.

The procedure for constructing the input pattern of an unknown acoustic event occurring at frame index $j$ is exactly similar. During the application phase, the input feature vector $\hat{\mathbf{x}}_j$ (which is normalized to have L2 norm equal to 1) is compared with all different class instances in order to find the class with the maximum fit as

$$I_j = argmax_{i,v} \left| \langle \mathbf{a}_{i,v}, \hat{\mathbf{x}}_j \rangle \right|, \quad (5)$$

where $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^H \mathbf{b}$ denotes the inner product between two vectors and $I_j$ carries the index (and optionally the instance index) of the selected class. In other words, we use a first-nearest neighbour (1-NN) classifier with inner product as the similarity measure.

## 4. EVALUATION

### 4.1 Description of the objects

We aim at providing results for three different objects; an old cassette-case made of recycled paper, a plastic bucket (originally used as a garbage bin) and an empty bottle of beer. From now on, we will refer to these object as the *box*, the *bucket* and the *bottle* respectively. The bucket and the box are excited with the fingers of both hands of the player whereas the bottle is excited with the help of a thin metallic rod.

For the box we have defined four different impact regions, using three out of the five available surfaces. Regions 1 and 2 are shown in Figure 1(a) while subfigure (b) depicts regions 2, 3 and 4. The box is placed in front of the user as in subfigure (c). It comes then very natural to excite region 1 by using only the fingers of the right hand while region 4 is optimal for the fingers of the left hand. Regions 2 and 3 can be easily struck with fingers from either hand.

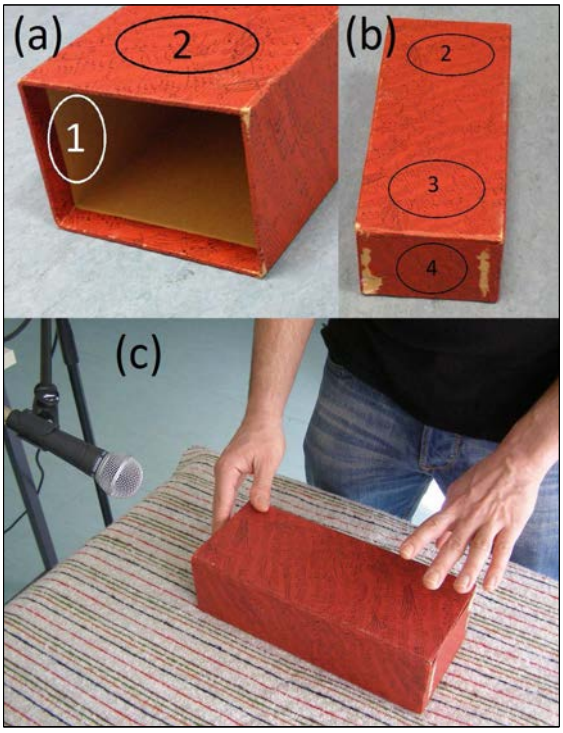


**Figure 1**. A box made of recycled paper. Impact regions 1 and 2 are shown in (a) while regions 2, 3 and 4 are shown in (b). The placement of the microphone with respect to the object and the general setting for playing the object is shown in (c).

For the bucket, four different impact regions are exploited, one on the vertical surface and three on the horizontal surface (see Figure 2(a)). The optimal location for the bucket is to place it upside-down in front of the user, as in subfigure (b). Again here, it comes natural to excite region 4 with the fingers of the left hand whereas for regions 1, 2 and 3 fingers from both hands may be used. For both the bucket and the paper-box, the different regions are excited by any of the index-finger or the middle-finger. The thumb, the ring-finger and the little-finger are not used for striking the object, but they are proved to be useful for supporting the object during performance (preventing it from unwanted displacements) and for stabilizing the positioning of the hands.

For the bottle we consider three different impact regions as shown in Figure 3(a). This setup exploits the smooth increment of the cross-section of the bottle along its main

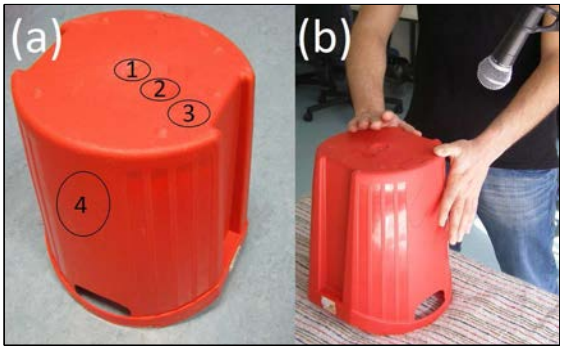


**Figure 2**. A bucket made of plastic. Impact regions 1, 2, 3 and 4 are shown in (a). The placement of the microphone and the general setting for playing the object is shown in (b).

axis. Two pieces of carton are used on either sides of the bottle in order to prevent unwanted displacements as shown in subfigure (b). In all cases, the objects are lying on a blanket which lies on the table. This was useful not only for stabilizing the objects but also for preventing the vibrations to be transmitted to the table.

### 4.2 Recording and training

Recordings took place in a relatively large room of the university (8 x 7 x 2.5 m). A cardioid dynamic microphone (Shure SM 58) plugged into an external USB sound card was used for acquiring the audio data during both the training and the testing phase. The sampling rate was set at 44100 but the audio data was downsampled at 22050 Hz for further processing.

The training data was automatically extracted from the corresponding audio files by using the onset detection algorithm; long audio files were segmented into multiple smaller files containing a single impact sound each. For each object and impact region, 35 to 50 instances were recorded. For the bucket and the box, strikes from both fingers and hands were recorded (when applicable) at each impact region. Also, for all objects and regions, we have tried to produce different intensity levels in order to cover a wide dynamic range.

In order to give an impression about the duration of the acoustic events involved in the classification task, we have plotted one representative waveform from each object in Figure 4. In general, we have observed that the acoustic energy drops below 40 dB within 0.2 s after the onset for all three objects.

### 4.3 Rhythmic patterns used for testing

In an attempt to decide upon the rhythmic patterns that would be used for testing the different objects, we focused on two important aspects; that all impact regions on each object should be excited a more or less equal number of times and that the testing sequences should include parts of non-trivial rhythmic complexity. Inspired by classical exercises for drummers, apart from a simple rhythmic pattern, the bucket and the box were excited with a series of

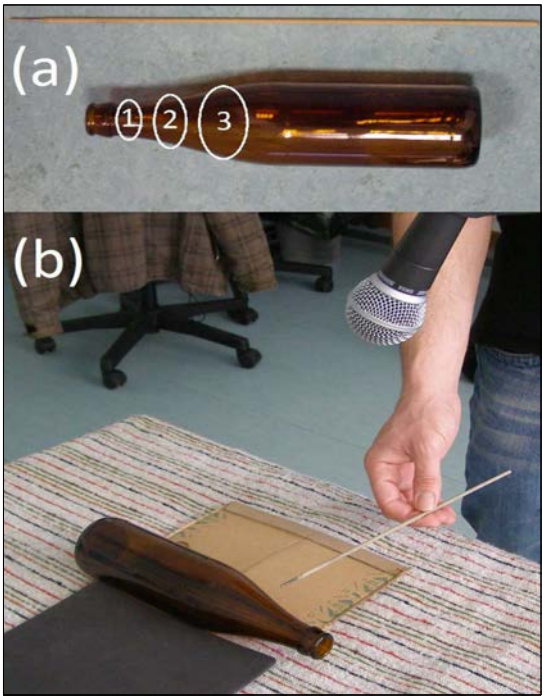


**Figure 3**. An empty bottle of beer. Impact regions 1, 2, 3 are shown in (a) together with the metallic rod that was used for hitting the object. The placement of the microphone and the general setting for playing the object is shown in (b).

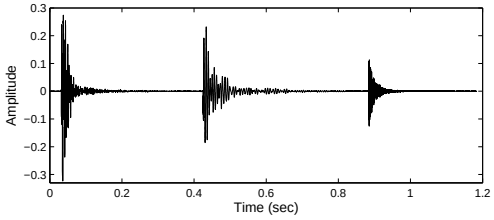


**Figure 4**. From left to right, one instance of a strike on the box, the bucket and the bottle.

"drum-rolls" and "double-strikes". Especially in the case of the "double-strikes", the distance between successive onsets was close to 30 ms and there was an evident overlap between adjacent events. For the bottle, apart from a simple rhythmic pattern we played "double-stroke-rolls", a classic technique which exploits the natural rebound of the drum-stick (in our case a metallic rod) in order to perform a second strike which rapidly follows the first one. Successive onsets as close as 70 ms apart were produced this way.

During the recording of the test audio files for each object, the positions of the object and the microphone were kept the same as during the recording of the training files so that the acoustic conditions are the same. About 500 events were recorded for each object and the events were manually labelled.

### 4.4 Onset detection performance

It was observed that the detection performance was stable for a wide range of different detection parameters for all three objects. However, some of the parameters appeared to have a significant impact on the classification performance. It was observed for example that increment of the hop-size $h$ led to a significant deterioration in the classification accuracy, especially for cases corresponding to small lengths of the analysis frame $t_{fr}$. This is not surprising, considering that increment of the hop-size quickly makes it comparable to the duration of the analysis frame $t_{fr}$. The uncertainty associated to the location of the onset causes the input feature vectors to be "misaligned" with respect to the training feature vectors. As a result, we may fail to observe a good fit and the event may be mistakenly assigned to the wrong class. Similar cases of misalignment may be observed when the threshold value $A_{tr}$ or the length of the window that is used for onset detection is different during the training phase and the testing or the application phase. As a general rule, we propose small hop-sizes and that the detection parameters in the testing phase are exactly the same as the ones used in the training phase.

The final values of the parameters associated to onset detection, which were the same for all three objects, are the following; a Hanning window of 3 ms duration was used with a hop-size of 16 samples (0.73 ms at 22050 Hz). The spectral energy measures $A$ and $B$ defined in subsection 3.1 were calculated over the frequency range of 800 to 6000 Hz. Thresholds $A_{tr}$ and $B_{tr}$ were set at 13 and .033 while $t_{ant}$ was set at 21 ms. The parameters for onset detection were exactly the same during both the training phase and the testing phase.

Overall, the onset detection algorithm was very accurate. Out of 1470 true percussive events, there was 1 missed onset and 9 false positives. In the current phase, we haven't taken any measures for treating false positives. They were simply disregarded during the calculation of the classification scores presented in the following section.

### 4.5 Classification performance

The frequency limits for the construction of the spectral feature vectors, $f_{min}$ and $f_{max}$ defined in section 3.2, were free parameters in the classification process. We do not have any sophisticated method to report for tuning these values, although they proved to be quite crucial for the overall performance. After a few trials, we decided to set these values to 0 and 1200 Hz for the box and the bucket and to 1000 and 5000 Hz for the bottle. Although the bottle had strong modal components above this frequency limit, we realized that there is no significant benefit by accounting for higher frequencies. To be noticed that, having kept a database with the original training instances, the spectral feature vectors corresponding to different STFT lengths and different frequency limits of $f_{min}$ and $f_{max}$ could be extracted immediately and classification scores were derived instantaneously for each combination of parameters.

Classification results are shown for lengths of the analysis frame of 3, 5, 7.5 and 21 ms in Table 1, assuming that a single object with a known identity is stroked at each time. The values outside and inside the parenthesis correspond to the case of complex and real feature vectors respectively. It can be seen that even with a 5 ms analysis frame, the classification performance is above 90% for all three objects. Observe that accounting for the phase of the STFT in the feature vectors brings a significant advantage in the case of the box and the bucket, especially at small values of $t_{fr}$. On the contrary, classification scores are a little better without phase information for the bottle.

| $t_{fr}$ | Box | Bucket | Bottle |
|---|---|---|---|
| | 0-1.2 kHz | 0-1.2 kHz | 1-5 kHz |
| 3 ms | 90% (79%) | 80% (65%) | 93% (93%) |
| 5 ms | 99% (89%) | 91% (86%) | 93% (95%) |
| 7.5 ms | 98% (96%) | 93% (88%) | 95% (97%) |
| 21 ms | 98% (94%) | 95% (94%) | 95% (99%) |

**Table 1**. Classification scores for each object in the single-object scenario. Values outside and inside parenthesis correspond to complex and real feature vectors respectively

It should be noticed that the small size of the feature vectors in combination with the low complexity of the nearest neighbour search makes the process ideal for a real-time application. Implemented in `Matlab` on a 3.4 GHz processor, the average computation time required for classifying a single event varied between 0.14 to 0.2 ms for the case of a 3 and a 21 ms length of the analysis frame respectively. This indicates that $t_{fr}$ is by far the most dominant factor determining the latency of the system, although an additional delay should be expected in accordance to the actual size of the audio buffer that would be used in the case of a real-time application.

We would also like to report results for the case of "multiple" objects, when the identity of the object that produced the event is not known and must be inferred from all 11 possible classes. In order to have a common basis for comparing among the three different objects, the feature vectors were here constructed as follows; we considered a wide frequency range from 0 to 5000 Hz for all three objects. The part of the feature vectors corresponding to frequencies from 0 to 1200 Hz was complex (magnitude and phase) while the remaining part was real (magnitude only). The test audio files and the onset detection parameters used were exactly the same as the ones used for the single object scenario. Classification scores are shown in Table 2.

| $t_{fr}$ | Box | Bucket | Bottle |
|---|---|---|---|
| | 0-5 kHz | 0-5 kHz | 0-5 kHz |
| 3 ms | 95% | 84% | 86% |
| 5 ms | 91% | 91% | 96% |
| 7.5 ms | 94% | 93% | 96% |
| 21 ms | 97% | 95% | 97% |

**Table 2**. Classification scores for each object in the multi-object scenario.

In general, one would expect a deterioration in the classification scores but we see that for some cases the results are improved. Observe for example that the classification scores for 3 ms are better in the case of the box and the bucket in comparison to Table 1. This is a consequence of adding more high-frequency information into the feature vectors. It should be mentioned that none of the events originating from the bottle were miss-classified as belonging to the box or the bucket and vice-versa. This proves that the deterioration observed for the bottle at 3 ms (-7%) is the result of adding low frequency information into the feature vectors. Nevertheless, this gives a 1% advantage at the longer analysis frame-lengths.

In several cases, events originating from the box were mistakenly assigned to the bucket, although the opposite case was not so common. At 5 ms for example, 56 out of the 632 events recorded from the box were assigned to the bucket and this confusion was the main reason for the score dropping at 91% from 99%. At higher frame-lengths this confusion became much more rare.

As expected, the average computation time required for classifying a single event was increased in the multi-object scenario. In the current approach the nearest neighbour is searched within all instances of all 11 classes. Even in this exhaustive approach, we may report computation times of 0.25 and 0.35 ms for the case of a 3 and a 21 ms analysis frame-length respectively.

## 5. PERSPECTIVE

The ability to infer the identity of the region of impact on a simple object may be exploited in order to turn the object into an accurate control interface with application in a variety of HCI systems. In this paper, detection and classification are still implemented offline, but the presented approach may be easily extended towards a stand-alone real-time application. It is one of our highest priorities to implement this step and to build a platform for using the detection and classification decisions as the input stream for controlling a real-time percussive synthesizer, turning thus the whole system into a low-cost real-time percussion controller.

In many aspects, the method used in this work is rather naive and we feel that, with moderate effort, both the speed and the accuracy of the system can be improved. We are currently investigating the use of different types of features and distance measures as well as techniques for reducing the number of class instances in each dictionary. We also foresee an interesting perspective in unifying the onset detection and classification process, exploiting thus the available acoustic models for discriminating between false and true onsets. An additional topic of concern is the possibility to acquire knowledge on-the-fly, i.e. to allow the online adaptation of the dictionaries, by selectively adding new impact patterns as they occur during the application phase, or by updating the already existing ones.

An additional research priority is to examine how well the process behaves under less ideal conditions and equipment than those in the presented experiments. How much does the process degrade when using lower quality sensors and recording formats, such as those found in most mobile devices? Also, how robust is the application to changes in the position of the object or the microphone and how is this related to the type of the feature vectors, the classification method and the adaptation method which is used?

While the pure artistic value of the natural sounds of the three objects used in this experiment is relatively poor in comparison to real percussive instruments, they prove to be advantageous as control interfaces, mainly due to the rapid decay that characterizes them. On the other hand, there is in our opinion still little effort focused in the topic of instantaneous classification of real percussive instruments. As already said, it is our intuition that acoustic structures originating from real percussive instruments will be a more challenging case for instantaneous classification tasks, because of the longer acoustic tails involved as well as because of the existence of simultaneous events (i.e hi-hats and bass-drum hits occurring at the same time). However, there is a large amount of tools which may assist in this case; source separation may be used in order to separate an attack segment from the tail of the previous hit and sensor arrays providing spatial information may assist further in both discriminating and separating the acoustic signals according to their locations or directions of arrival. A natural consequence of the last ideas is to treat the classification task in terms of a Multiple-Input Multiple-Output (MIMO) problem, where information from multiple acoustic sensors is exploited in order to discriminate between multiple classes. This scenario is also particularly appealing to the classical multichannel setup which is used in pop and rock music for sound reinforcement and recording applications, where many microphones are distributed around the drumset. In this direction, the application may be seen as a non-invasive sensing solution for replacing classical drum triggers, whose installation is elaborate and which some times affect the acoustic behavior of the instrument.

Instantaneous detection and classification of percussive events is to our opinion a prerequisite towards a more fascinating and interdisciplinary research topic which considers the extension of the capabilities of classical percussive instruments and their transformation into *hyper-instruments*. Hyper-instruments is a concept developed by Tod Machover [16] in which real physical objects are fitted with electronic sensors as gestural acquisition devices. The gestures are transformed into control messages to a computer for producing a sound to accompany the real physical instrument or for performing some other predefined action. Examples of these ideas in the family of percussive instruments may be found in the works of Mann et al. [17] and Trail et al. [18]. They both illustrate a high-level gesture control interface, but their implementation relies on the use of special sensor devices (position sensors, cameras, radars etc.) which are usually not in the possession of common musicians. In this regard, the use of solely acoustic sensors as the gesture acquisition device carries the potential for reducing the cost of implementation and for achieving the vision of a *generic* solution, which would enable the users to train the system to respond to their already disposable instruments and audio equipment.

## 6. CONCLUSIONS

Inferring percussive gestures from acoustic data in real-time may be seen as the core process for designing many fascinating applications related to musical control interfaces and HCI systems in general. In this paper, we have used a simple instance-based classification technique in order to train the system to recognize the differences in the resonant behavior of one or more objects as these objects are struck by the user at different locations. Simple spectral features of the monophonic acoustic signal provide sufficient discriminatory information for achieving classification rates above 90%, with a system response of 5 ms or even less.

The primary author occasionally uses the three objects presented in this paper for programming drum tracks which reproduce the rhythmic section in his personal music compositions. The onset detection and classification results are transformed into a MIDI file which is then imported into a Digital Audio Workstation for controlling a sample-based percussive synthesizer. He finds it much more natural to interact with these objects than with a piano-like MIDI controller that he has in his possession. The box and the bucket are very convenient for programming classical membranophones such as bass-drums, snare-drums and toms, while the bottle is very suitable for programming hi-hats, rides and cymbals in general.

### Acknowledgments

This research has been co-financed by the European Union and Greek national funds through the National Strategic Reference Framework (NSRF), Research Funding Program: "Cooperation-2011", Project "SeNSE".

## 7. REFERENCES

[1] P. Herrera, A. Dehamel, and F. Gouyon, "Automatic labelling of unpitched percussion sounds," in *Proc. 114th Conv. Audio Engineerig Society*, Amsterdam, 2003, pp. 1–14.

[2] V. Sandvold, F. Gouyon, and P. Herrera, "Percussion classification in polyphonic audio recordings using localized sound models," in *Proc. Int. Conf. Music Information Retrieval*, Barcelona, 2004, pp. 537–540.

[3] K. Yoshii, M. Goto, and H. Okuno, "Adamast: A drum sound recognizer based on adaptation and matching of spectrogram templates," in *Proc. Int. Conf. Music Information Retrieval*, Barcelona, 2004, pp. 184–191.

[4] F. Gouyon, F. Pachet, and O. Delerue, "On the use of zero-crossing rate for an application of classification of percussive sounds," in *Proc. COST G-6 Conference on Digital Audio Effects (DAFX-00)*, 2000.

[5] A. Zils, F. Pachet, O. Delerue, and F. Gouyon, "Automatic transcription of drum tracks from polyphonic music signals," in *Proc. 2nd Int. Conf. on WEB delivering of Music (WEDELMUSIC'02)*, 2002, pp. 179–183.

[6] K. Tanghe, S. Degroeve, and B. D. Baets, "An algorithm for detecting and labeling drum events in polyphonic music," in *Proc. First Annual Music Information Retrieval Evaluation Exchange (MIREX)*, 2005.

[7] M. Puckette, T. Apel, and D. Zicarelli, "Real-time audio analysis tools for pd and msp," in *Proc. Int. Computer Music Conf.*, Ann Arbor, 1998, pp. 109–112.

[8] G. Weinberg and S. Driscoll, "Toward robotic musicianship," *J. Computer Music*, vol. 30, no. 4, pp. 28–45, 2006.

[9] A. Jylhä, I. Ekman, C. Erkut, and K. Tahiroglu, "ipalmas-an interactive flamenco rhythm machine," Glaskow, 2009, pp. 69–76.

[10] U. Şimşekli, A. Jylhä, C. Erkut, and T. Cemgil, "Real-time recognition of percussive sounds by a model-based method," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, pp. 1–14, 2011.

[11] E. Battenberg, "Techniques for machine understanding of live drum performances," Ph.D. dissertation, EECS Department, University of California, Berkeley, Dec 2012. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-250.html

[12] B. Zamborlin, F. Bevilacqua, M. Gillies, and M. D'Inverno, "Fluid gesture intgeraction design: Applications of continuous recognition for the design of modern gestural interfaces," *ACM Transactions on Interactive Intelligent Systems*, vol. 3(5), 2014, article 22.

[13] A. Jylha and C. Erkut, "A hand clap interface for sonic interaction with the computer," in *Proc. Human Factors in Computing Systems*, Boston, 2009, pp. 3175–3180.

[14] S. Vesa and T. Lokki, "An eyes-free user interface controlled by finger snaps," in *Proc. 8th Int. Conf. Digital Audio Effects*, Madrid, 2005, pp. 262–265.

[15] H. Tan, Y. Zhu, L. Chainsorn, and S. Rahardja, "Audio onset detection using energy-based and pitch-based processing," in *Proc. 2010 IEEE Int. Symposium on Circuits and Systems*, Paris, 2010, pp. 3689–3692.

[16] T. Machover, *CyberArts: Exploring Art and Technology*. Miller Freeman, 1991.

[17] S. Mann and R. Janzen, "The xyolin, a 10-octave continuous-pitch xylophone and other existemological instruments," in *Proc. Int. Comp. Music Conf.*, 2012.

[18] S. Trail, M. Dean, T. Tavares, G. Odowichuk, and P. Driessen, "Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect," in *New Interfaces for Musical Expression*, Ann Arbour, 2012.