ORIGINAL RESEARCH
published: 10 July 2018
doi: 10.3389/fict.2018.00016

Real-Time Hit Classification in a Smart Cajón

Luca Turchet, Andrew McPherson and Mathieu Barthet*

Centre for Digital Music, School of Electronic Engineering and Computer Science, Queen Mary University of London, London, United Kingdom

Smart musical instruments are a class of IoT devices for music making, which encompass embedded intelligence as well as wireless connectivity. In previous work, we established design requirements for a novel smart musical instrument, a smart cajón, following a user-centered approach. This paper describes the implementation and technical evaluation of the designed component of the smart cajón related to hit classification and repurposing. A conventional acoustic cajón was enhanced with sensors to classify position of the hit and the gesture that produced it. The instrument was equipped with five piezo pickups attached to the internal panels and a condenser microphone located inside. The developed sound engine leveraged digital signal processing, sensor fusion, and machine learning techniques to classify the position, dynamics, and timbre of each hit. The techniques were devised and implemented to achieve low latency between action and the electronically-generated sounds, as well as keep computational efficiency high. The system was tuned to classify two main cajón playing techniques at different locations and we conducted evaluations using over 2,000 hits performed by two professional players. We first assessed the classification performance when training and testing data related to recordings from the same player. In this configuration, classification accuracies of 100% were obtained for hit detection and location. Accuracies of over 90% were obtained when classifying timbres produced by the two playing techniques. We then assessed the classifier in a cross-player configuration (training and testing were performed using recordings from different players). Results indicated that while hit location scales relatively well across different players, gesture identification requires that the involved classifiers are trained specifically for each musician.

Keywords: smart musical instruments, internet of musical things, embedded systems, sensor fusion, gesture classification, machine learning

OPEN ACCESS

Edited by:
Inmaculada Hernaez Rioja,
University of the Basque Country (UPV/EHU), Spain

Reviewed by:
Kening Zhu,
City University of Hong Kong, Hong Kong
Jiri Pribil,
Institute of Measurement Science, Slovak Academy of Sciences, Slovakia

****Correspondence:***
Luca Turchet
luca.turchet@qmul.ac.uk

Specialty section:
This article was submitted to Human-Media Interaction, a section of the journal Frontiers in ICT

Received: *04 April 2018*
Accepted: *19 June 2018*
Published: *10 July 2018*

Citation:
Turchet L, McPherson A and Barthet M (2018) Real-Time Hit Classification in a Smart Cajón. Front. ICT 5:16. doi: 10.3389/fict.2018.00016

1. INTRODUCTION

The cajón is a cuboid-shaped percussion instrument originally from Peru, which has recently become widely used worldwide and across various musical genres (Tompkins, 2007). The large diffusion of such an instrument may be attributable to its ability to produce a multitude of percussive sounds by hitting one or more of its sides, as well as to its portability. These aspects provide a viable and more cost effective alternative to the drumset.

A typical cajón consists of a hollow wooden box with a resonant chamber, which has a hole in the back wall for producing bass tones. A supplemental rattle device consisting of metal strings or snares is usually attached to the interior side of the front panel, to produce a snare-like sound especially when the top part of the front panel is struck. Typically, musicians who play the cajón sit on the top side and slap the front and side panels with their hands. When struck, each panel vibrates causing the displacement of air to produce sound (Ludwigsen, 2017). Striking a cajón panel in different places (e.g., high or low parts, the corners, or the central portion), and with different techniques (e.g., open or closed hands, fingers, palms, knuckles, or fingernails) can produce a variety of distinct percussive sounds. Like the vast majority of conventional acoustic instruments, the cajón affords high intimacy of control as it accommodates a wide variety of performance gestures and offers tactile feedback from the interaction of the hands with the panels which is coherent with the produced sound and in response to the performed gesture. This high level of "control intimacy" (Moore, 1988) is typically lost in most common percussive digital music interfaces, which are capable of tracking only the time and intensity of a hit, and in some cases different hit locations along a same surface (for reviews of percussive digital controllers see Aimi, 2007; Jathal, 2017).

In the past few years, a handful of acoustic cajones embedding electronic components have been proposed by different companies to extend the sonic possibilities of the instrument in its original version. Examples include Roland's Electronic Cajón[1], De Gregorio's Cajón Centaur[2], and Duende's Magik Cajón[3]. These instruments are equipped with sensors that can detect players' hits on specific regions of the instrument and trigger in response audio samples from different percussive instruments. In the De Gregorio's and Duende's cajones the sensors consist of pads placed on one of the side panels, and thus the electronic sounds resulting from hitting on them are unrelated to the acoustic sound and tactile sensation a musician would experience by interacting with the actual wood. Roland's instrument leverages another approach, by placing sensors underneath the wood on the middle-bottom and top central zones of the front panel. However, to date such instruments are not capable of detecting with high accuracy the location of a hit on all panels musicians interact with (i.e., the front, left, and right panels) nor capturing the richness of the hand-panel interactions in terms of amplitude and timbral nuances. Therefore, the rendering of the simulated percussive instrument sounds cannot be informed by a plethora of data describing how and where the hit was performed. Consequently, as stated by some of the cajón players we worked with in a previous study (Turchet et al., 2018), the quality of the gesture-to-electronic sound interaction is largely impoverished compared to the level of control intimacy of the acoustic cajón. Enabling subtle interaction and nuances in electronically-controlled cajones is deemed crucial by players (Turchet et al., 2018) and is the focus of the current study.

Electronic cajones currently available in the market belong to the family of the so-called "augmented instruments" (Miranda and Wanderley, 2006), which are familiar instruments whose musical capabilities are enhanced with sensors or actuators. Recently, a new family of musical instruments that builds upon the augmented instruments concept has been proposed, that of the "smart instruments" (Turchet et al., 2016). This is a class of Internet of Musical Things devices for music making, which encompasses embedded intelligence responsible for handling sensors and audio processing, as well as wireless connectivity. Internet of Musical Things (IoMusT) refers to interfaces, protocols and music-related data in an ecosystem of interoperable devices dedicated to the production and/or reception of music, which can lead to novel forms of interactions between performers and audiences (Turchet et al., 2017).

In previous work, we established design requirements for a smart cajón, following a user-centred approach (Turchet et al., 2018). Specifically, we conducted individual co-design sessions with five professional cajón players, which resulted in crafting and evaluating a prototype. Such prototype consisted of an acoustic cajón enhanced with a two-head contact microphone attached on the front panel; a sensor interface tracking various performing gestures mapped to different sound processing algorithms; a system of actuators delivering tactile feedback in response to messages sent from connected devices; a loudspeaker for delivering electronically-generated sounds, used along with a smartphone placed on top of it providing a touchscreen; an embedded computational unit responsible for wireless connectivity as well as processing of audio, sensors, actuators, and visual display. The sound engine was devised to track two zones of the front panel, the central top and the middle-bottom one. This was achieved by applying a discriminative threshold on the value of the spectral centroid extracted in real-time, leveraging the fact that hits on the two zones are typically characterized by different spectra (i.e., richer in high frequencies for hits on the central top part). However, such a method proved to be non optimal as in various cases the two zones were wrongly tracked.

In this paper, we investigate a more robust technique overcoming the limitations of our previous method based on spectral centroid. We focus on the design and development of a system capable of addressing some of the suggestions made by professional players that assessed the smart cajón prototype reported in Turchet et al. (2018). Those players highlighted the following areas of improvements related to expressive control: (i) track hits on more areas on the front panel, specifically the two top corners; (ii) track hits on the side panels; (iii) improve the sound quality of the triggered samples to render more adequately the type of gesture used (especially in terms of dynamics and timbral nuances); (iv) automatically transcribe the played patterns into a digital score. The present paper describes the implementation and technical evaluation of the designed component of the smart cajón related to the classification of the hit position and gesture that produced it, as well as its repurposing into a transcribed score and sound samples.

The remainder of the paper is organized as follows. Section 2 reviews the literature on which our work is grounded. Section 3 describes the design requirements, while sections 3.1 and 4 present our implementation at hardware and software levels. Section 5 presents the results of the technical evaluation performed with cajón players, while section 6 proposes a discussion and conclusion.

[1] www.roland.com/us/products/el_cajon_ec-10

[2] www.cajondg.com/product/cajon-centaur/?lang=en

[3] https://www.youtube.com/watch?v=zZO1y0aVnXQ

2. RELATED WORKS

2.1. Real-Time Music Information Retrieval

Music Information Retrieval (MIR) is an interdisciplinary research field focusing on retrieving information from music (Burgoyne et al., 2016). One of the main goals of this field is the development of effective methods capable of extracting temporal and spectral aspects of a music signal, especially for automatic music transcription purposes (Benetos et al., 2013). To date, the majority of MIR research has focused on offline methods analyzing audio files. Nevertheless, different techniques have also been developed for real-time scenarios, especially for retrieving information from the audio signal of a single musical instrument. In the context of percussive instruments, Miron et al. proposed a real-time drum transcription system available for the two real-time programming languages Pure Data (Pd) and Max/MSP (Miron et al., 2013a,b).

One of the crucial steps in the information retrieval process is the onset detection of musical events, that is the instant at which a pitched or unpitched musical sound starts. A plethora of techniques have been developed for this purpose (see e.g., Bello et al., 2005; Dixon, 2006; Stowell and Plumbley, 2007), including those that rely on the fusion of various methods (Tian et al., 2014). Real-time implementations of some of such techniques have been made available in the *aubio* open source library for Pd currently maintained by Paul Brossier[4].

After the detection of an onset it is possible to retrieve information from the corresponding musical event, such as for instance its timbre, which is a crucial aspect for expressivity (Barthet et al., 2010). Various techniques are available for this purpose, some of which are more specific for percussive sounds and for real-time contexts. In Brent (2009), the author describes a set of temporal, spectral, and cepstral features that are relevant and useful for percussive timbre identification, and which can be computed in real-time. These are included in the *timbreID* library for Pd[5] (Brent, 2010). Such library, besides providing efficient implementations of a set of low-level temporal, spectral, and cepstral feature-extraction techniques, also integrates a real-time classifier based on machine learning algorithms, which takes in input vectors of extracted audio features.

2.2. Musicians' Gestures Tracking

A considerable amount of research in the fields of MIR and NIME has focused on the automatic sensing of the gestures of the performers interacting with their instruments (Jensenius and Wanderley, 2010). There are two main approaches to sensing gestures performed on musical instruments: sensor augmentation and indirect acquisition (Driessen and Tzanetakis, 2018). Sensor augmentation entails the modification of the conventional instrument by adding sensor technology to it (e.g., force sensing resistors, accelerometers) to measure various aspects of the performers' gestures. Various augmented instruments have been crafted for this purpose, including the percussive ones (e.g., Kapur et al., 2004; Young and Fujinaga, 2004; Michalakos, 2012). In indirect acquisition the only sensor utilized is a microphone that captures the sound produced by the instrument. Examples of indirect acquisition systems developed for percussive instruments gesture extraction are reported in Gouyon and Herrera (2001), Tindale et al. (2004, 2005), and Jathal (2017). The algorithms proposed in those works rely on signal processing, possibly followed by machine learning.

Both sensor augmentation and indirect acquisition have advantages and disadvantages. While sensor augmentations provide relatively straightforward and reliable measurements, they require invasive modification of the instrument, which is frequently undesirable. On the other hand, indirect acquisition is non-invasive but requires the use of sophisticated signal processing and possibly machine learning algorithms to extract from the audio signal the information relevant to classify a gesture. This requires high development efforts and in the presence of machine learning techniques, time consuming manual labeling processes (for supervised learning) and model training are necessary. Moreover, indirect acquisition may not achieve the same performance accuracy as direct sensors, and audio feature extraction typically adds latency. Tindale et al. proposed a sensor fusion approach to exploit the advantages of the two methods, by developing an efficient and effective system that used direct sensing via sensors to train a machine learning model for indirect acquisition (Tindale et al., 2011).

Sensor fusion refers to a process by which data from different sensors are merged to compute something more than could be determined by any one sensor alone (Liggins et al., 2017). Such a technique has been utilized in the context of musical instruments. For instance, MacRitchie and McPherson proposed a method to integrate streams of information coming from a markers-tracking camera and capacitive touch sensors attached to the keys of a piano (MacRitchie and McPherson, 2015). The method was successful in accurately analyzing pianists' fingers movements. Odowichuk et al. combined data extracted from a Kinect motion sensing device (by means of computer vision techniques) with data of a the Radio-drum (an electromagnetic capacitive 3D input device) in order to infer gestures of a performer playing a vibraphone (Odowichuk et al., 2011). A complicating factor in these systems is that the sources of information are sensed by different devices, each of which has its own clock and its own sample/frame rate. This may lead to issues with temporal and spatial alignment of the collected data. In a smart instrument instead, all sources of information may be processed with the same clock and in some cases even with the same sampling rate. Along these lines, Pardue and McPherson presented an approach to violin pitch tracking combining audio and position sensor data, showing that the combination of the two sources of information outperformed audio-only methods (Pardue et al., 2015).

3. DESIGN

To progress the state of the art compared to our previous prototype as well as existing commercial solutions, we designed a system with the following requirements for tracking and

[4] Available online at: www.aubio.org

[5] Available online at: www.williambrent.com

repurposing the hits. In reference to **Figure 1**, the system was designed to track:

- hits on 3 zones in the front panel: bottom (FB), top-left (FTL), top-right (FTR);
- hits on the left (L) and right (R) panels;
- hits with a wide range of dynamics in each of the zones above;
- hits with a high temporal resolution;
- two different types of gestures associated to hits in each of the zones;
- different timbral nuances for each zone and each gesture.

The requirements for repurposing the hits were:

- to map each of the five zones (i.e., FB, FTL, FTR, L, and R) to a different sound sample;
- to map each gesture detected within a same zone to a different sound sample, to render the different gestures;
- to apply different equalizations for each gesture detected within a same zone, to render timbral nuances;
- to map the dynamics of each detected hit to the dynamics of the triggered sample;
- to automatically create a score.

The performance requirements for the overall system were:

- to deliver with imperceivable latency the electronically-generated sounds in response to hits;
- to keep as low as possible the computational load.

The design process was informed not only by the results of the study reported in Turchet et al. (2018), but also by consultations with three professional cajón players who were involved in co-design and testing sessions before the final evaluation.

In general, control intimacy was adopted as a design criterion (Moore, 1988). Therefore, we focused on the most crucial aspects for the interaction such as hit detection, hit position and gesture classification, rendering of timbral nuances, and latency.

Notably, the concept of zone described here is just a simplification. Indeed, what really matters for a player is the match between the generated sound and the intention to generate it. For instance a bass sound, that is a sound richer in low frequencies, might be produced not only by playing exclusively in the FB zone, but also by playing at the same time on the FTR and FB zones (see **Figure 2**). The capability of producing specific timbres largely varies with the playing technique of a performer and his/her hand dimensions. Our system was designed to cope with this situation, taking into account the timbre of the sound produced. For convenience's sake, in the reminder of the article we will use BAS to indicate a zone that corresponds to a hit producing the bass sound (which typically comprises, but it is not limited to, the FB zone).

In this study we focused on the detection of two types of gestures utilized by cajón players: *slap hit* and *tap hit*. These gestures generate respectively a sharp sound and a more muted sound. Generally, slap hits are accomplished by slapping the instrument with the open hand and using the finger pads, while tap hits are produced by using the fingertips or having a more closed hand (see **Figure 2**). The detection of only these two hits was motivated by the fact that not only they are the ones most widely used, but also to keep the accuracy of hit detection high and the design of the related algorithms efficient. Other less common techniques such as those involving knuckles or nails will be investigated in future research.

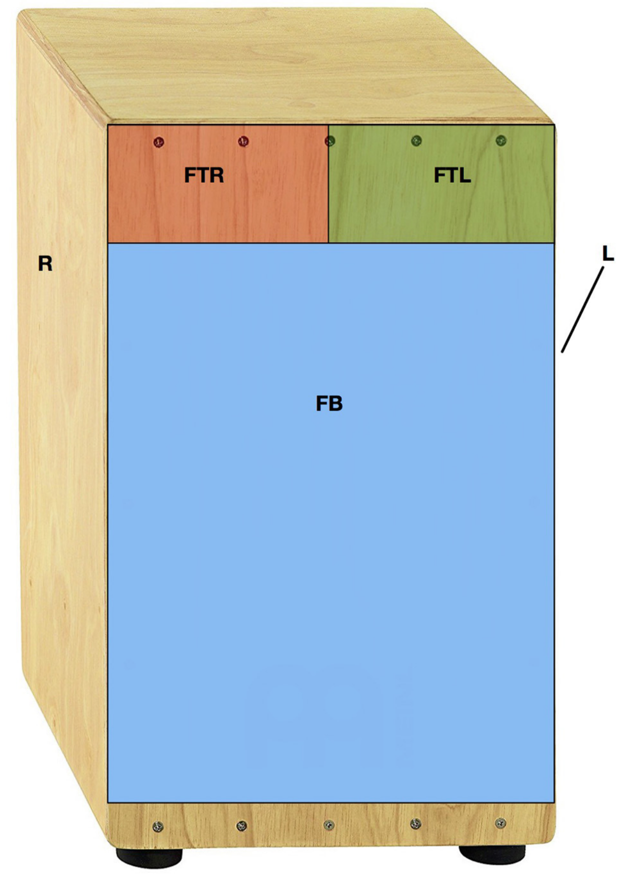

FIGURE 1 | Illustration of the zones of the instrument tracked by the system.

As far as the action-to-sound latency is concerned we targeted a limit of 20 ms, despite a generally accepted limit for latency in digital musical instrument is 10 ms (Finney, 1997; McPherson et al., 2016). This was motivated by the results of the evaluation of the smart cajón prototype reported in Turchet et al. (2018) were we found that none of the participants could perceive the measured average latency of 20 ms, likely because of a masking effect in the attack of the acoustic sound that superimposes over the digital one. Considering this and a tradeoff with accuracy, as well as with computational efficiency using low cost commercially available equipment, our targeted latency constraint between the actual hit performed on the instrument and the electronically-generated sound was set to 28 ms. Nevertheless, in our design we also considered that cutting edge technologies available in the music industry would ensure a latency of 22 ms (e.g., MIND Music Labs' ELK operating system for embedded platforms[6]).

[6] https://www.mindmusiclabs.com/elk/

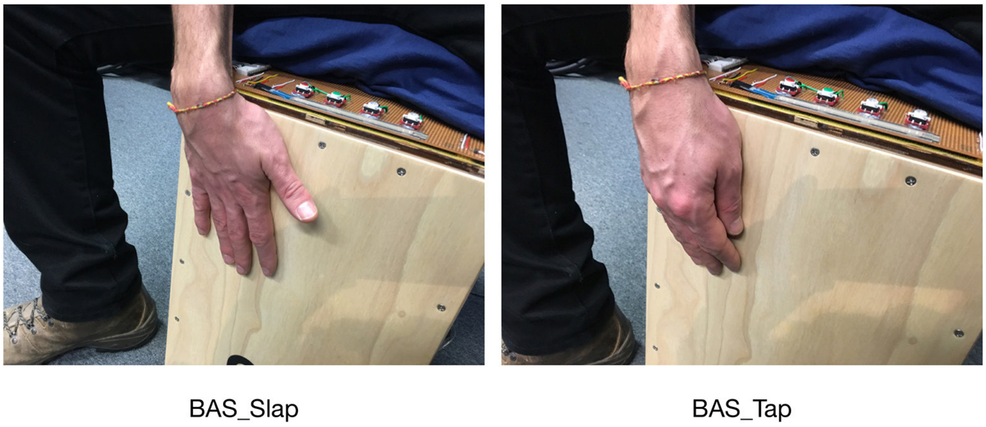

FIGURE 2 | Pictures of the slap and tap hits produced by a player's hand performing a sound related to the BAS zone.

The next two sections detail the hardware and software components of the prototype built according to the design requirements.

3.1. Prototype's Hardware

The prototype consisted of a conventional acoustic cajón, smartified with different hardware components. The computational unit consisted of a Bela board for low-latency audio processing (based on a Beaglebone Black board) (McPherson and Zappi, 2015). Bela was extended with a shield to obtain 10 audio inputs. Wireless connectivity was accomplished by means of a small wireless router (TL-WR902AC by TP-Link), which featured the IEEE 802.11ac Wi-Fi standard. Sound delivery was accomplished by a loudspeaker (Monitor Supreme Center 250 by Magnat) with small pre-amplifier (SA-36A Pro HIFI Digital Amplifier by SMSL). Power was supplied externally using AC power plugs.

To detect the acoustic sound produced by the instrument in response to a hit we used a pickup system composed by five piezoelectric microphones (Big Shot by K&K), as well as a condenser microphone. The piezoelectric microphones were attached to the instrument's internal panels by means of blu-tack adhesive, in correspondence to the five zones indicated in **Figure 1**. The reason for using a pickup system was motivated by the fact that a pickup relatively far from another is capable of producing temporally and dynamically different pickup signals for a same hit. Such differences could be exploited to detect the zone of the instrument in which the hit was produced. Critical to the use of this pickup system is the positioning of the each piezo as well as the hardware adjustment of the level of the corresponding input signal (by means of potentiometers) in order to limit as much as possible cross-talking effects. Specifically the piezo pickup placed on zone FB needed to have a much lower input gain than the other pickups due to the fact that it is attached to the more resonant part of the instrument.

The condenser microphone utilized was the Beta 91A by Shure. It is a preamplified, flat shaped, half-cardioid microphone with a wide and accurate dynamics tracking, as well as a tailored frequency response, which is designed specifically for kick drums and other bass instruments. It is frequently involved in professional contexts, such as recording and live performances, for miking the cajón: its placement inside the instrument allows for an optimal capture of the acoustic sound and also limits the interference from external sound sources (e.g., other musical instruments playing on stage). In our prototype such a boundary microphone was attached, by means of velcro, to a foam sealed to the rear part of the bottom side, to prevent the microphone to move around inside. Being a condenser microphone made necessary the use of a supplier of 48 V phantom power (we selected the PS400 model by Behringer for a tradeoff between small size and quality).

The reason for using these two miking apparati was due to their different and complementary tracking capabilities, which made them ideal candidates for the adoption of sensor fusion techniques. The piezo pickups, even when their signals are merged together, are unable to capture with the same level of accuracy than a condenser microphone all the timbral and dynamic nuances of the produced hits (this is especially true in our system due to the signal conditioning performed to achieve optimal spatial tracking of the hits, see section 4.2). On the other hand, the condenser microphone alone may be less well suited for the task of detecting the location of each hit.

4. PROTOTYPE'S SOFTWARE

A sound engine responsible for microphones processing and sound production was developed in Pd, running on the Linux operating system which comes with Bela. The audio buffer was set to 128 samples for efficiency of block-based processing, which led to an estimated round-trip latency of 6.7 ms (McPherson et al., 2016).

Data reception and forwarding over Wi-Fi were achieved using Open Sound Control (OSC) over the User Datagram Protocol. Following the recommendations reported in Mitchell et al. (2014) to optimize the components of a Wi-Fi system for live performance scenarios, in order to reduce latency and increase throughput, the router was configured in access point mode, security was disabled, and support was limited to theIEEE 802.11ac standard only. The wireless communication with a laptop allowed for the real-time monitoring and control of the status of the sound engine (e.g., for tuning the parameters of the detection algorithms and to start/stop recordings).

Figure 3 shows a block diagram of the overall process of triggering sound samples on the basis of the input signals, which was accomplished by the sound engine. Such a process leveraged digital signal processing, sensor fusion, and machine-learning techniques to classify the position, dynamics, and timbre of each hit. These techniques were devised and implemented to achieve low latency between action and the electronically-generated sounds, as well as keep as low as possible the overall computational load.

Specifically, the sound engine comprised eight modules:

1. **Onset Detection:** For each piezo pickup we detected the presence of an onset and calculated the corresponding peak in the signal.
2. **Hit Detection:** This module detects the presence of a hit by selecting the first of the onsets detected from the five piezo pickups within a certain checking period and after a certain refractory period.
3. **Hit Localization:** This module is responsible for detecting in which part of the instrument the hit was produced.
4. **Features Extraction:** This module computes algorithms to extract temporal and spectral features from the audio signal captured by the condenser microphone.
5. **Gesture Classification:** The spectral features computed in the Features Extraction module as well as the location of a hit are used as input for a gesture classifier based on supervised learning.
6. **Hit Classification and Automatic Score Transcription:** The type of hit is classified on the basis of the information about position and type of gesture, which are received from the hit localization and gesture classification modules. This module also implements a score transcription.
7. **Sample Selection and Parameters-to-Sound Mapping, and Triggering:** This module selects among a set of possible choices, one sound sample. The selection process is informed by the labels assigned to the hit by the Hit Classification module as well as by the extracted spectral information. The module is also responsible for assigning a volume to the sample based on the input signal's amplitude calculated by the Features Extraction module. Finally, the module triggers the playback of the selected sample.

The next subsections provide details about each of the eight modules.

4.1. Real-Time Onset Detection

We proposed an onset detection algorithm based on a novel approach combining time- and spectrum-based techniques (Turchet, 2018). Such onset detector was applied to each of the signals captured by the five pickups.

Typically to detect efficiently an onset using spectral methods at least 5.8 ms are needed after the occurrence of the peak of the

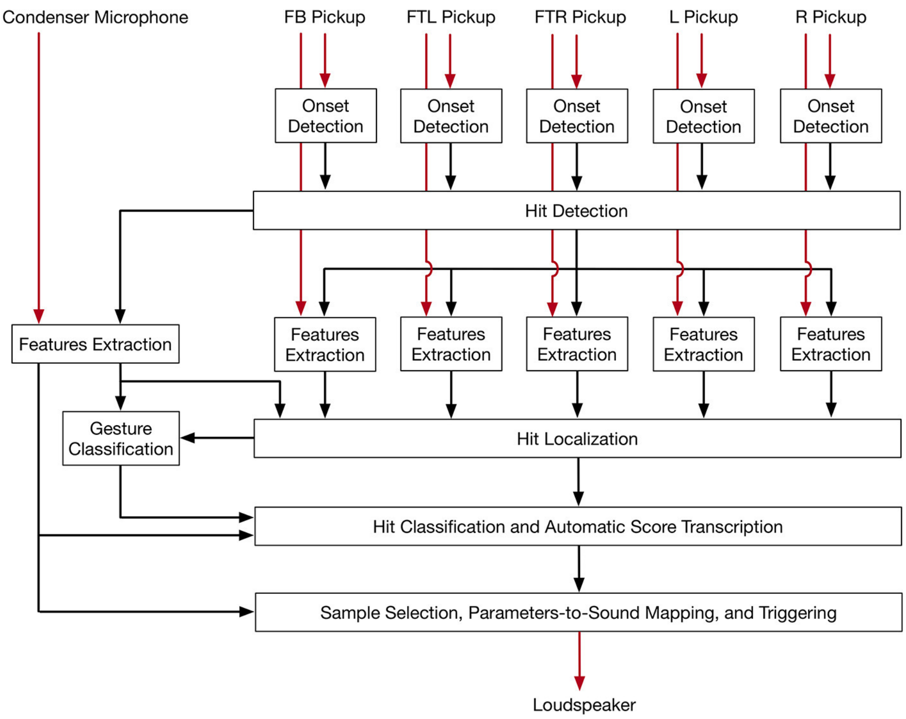

FIGURE 3 | Block diagram of the overall process for real-time hit detection, classification, and repurposing. In red the audio signals, in black the control data.

involved onset detection function (ODF), considering a window size of 256 samples for the Short Time Fourier Transform and a sampling rate of 44.1 kHz. For such methods the time between the actual onset and the reported onset is unpredictable and may largely vary according to the type of percussive sound. This is due to the fact that these methods are based on the identification of the ODF peak, and not on the actual initial moment of the hit. Specifically, we empirically found that for some hits the delay could even amount to 20 ms. This aspect was unacceptable in our application where it was fundamental to detect the exact moment in which a hit happened in order to perform the analyses on the first portion of the sound, and use the results of such analyses to classify different sounds related to different hits. On the other hand, our initial experimentations suggested that methods based on temporal features may have a higher degree of accuracy in detecting the initial moment when a hit is generated. Nevertheless, onset detection methods based on the spectral content may be less prone to false positives compared to methods based on temporal features if their parameters are appropriately tuned.

The onset detection method here proposed takes advantage of the strengths of the two approaches. Specifically, a time-based technique capable of detecting the initial moment of a hit but more sensitive to spurious detections, was used in parallel with a linear combination of two spectrum-based techniques, which are more robust to spurious detections (when appropriately tuned) but have higher and variable delay. We used two complementary spectral methods in parallel in order to improve accuracy in the case in which one of the two failed in detecting an onset. The idea beyond this mixed approach was not only to detect exclusively a single onset per hit and with minimal delay after the initial moment of contact of the hand with the wood, but also to ensure perfect tracking and a high temporal resolution in tracking two subsequent hits. We set such resolution to 30 ms since this is the temporal resolution of the human hearing system to distinguish two sequential complex tones (Moore, 2012).

We also investigated the case of applying the proposed onset detector only on the signal captured by the condenser microphone, but we found out that directly processing the signals of the pickups led to slightly better performances in terms of accuracy and delay. This is due to the fact that the pickups are capable of providing signals more defined and sharper, which makes the onsets easier to detect. However, despite its increased accuracy, this method is also much more expensive computationally than only analyzing the single signal from the condenser microphone.

4.1.1. Time-Based Onset Detection Technique

The time-based technique here proposed consisted of a modification of the approaches to onset detection described in Brossier et al. (2004) and Bello et al. (2005). It must be specified that this technique only provides as output an onset timing, not the associated peak amplitude. A peak-picking algorithm is instead performed in the Features Extraction module, on the original signal, not on the computed onset detection function (ODF) as normally happens for other onset detectors present in the literature.

We computed an ODF as follows. Firstly, we squared each sample above zero. This method discarded the negative part of the signal, which was useful to limit spurious detections after the attack. Our in-depth analysis on the original waveforms showed that such a method affected only minimally the accuracy time of the onset detection. The resulting signal underwent a smoothing process accomplished by a lowpass filter with cutoff frequency at 15 Hz. This was followed by the calculation of the first derivative and again the application of a lowpass filter with cutoff frequency at 15 Hz.

Subsequently, a dynamic threshold (capable of compensating for pronounced amplitude changes in the signal profile) was subtracted from the signal. We utilized a threshold consisting of the weighted median and mean of a section of the signal centered around the current sample n:

$$\delta(n) = \lambda \cdot median(D[n_m]) + \alpha \cdot mean(D[n_m]) \quad (1)$$

with $n_m \in [m - a, m + b]$ where the section $D[n_m]$ contains a samples before m and b after, and where λ and α are positive weighting factors. In our implementation we used a section of 64 samples with $a = 62$ and $b = 2$, as well as $\lambda = 0.5$ and $\alpha = 0.5$. For the purpose of correctly calculating the median and the mean around the current sample, the pre-thresholded signal was delayed of 2 samples before being subtracted from the threshold. The real-time implementation of the median was accomplished by a Pd object performing the technique reported in Herzog (2013).

The detection of an onset was finally accomplished by considering the first sample n of the ODF satisfying the condition:

$$n > \delta(n) \quad \& \quad \delta(n) > \beta \quad (2)$$

where β is a positive constant. The reason for using the fixed threshold β on the dynamic threshold rather than setting it on the actual detection function was due to the fact that it provided better detection performances. A refractory period of 30 ms was applied after such detection to discard false positives.

4.1.2. Spectrum-Based Onset Detection Techniques

Various algorithms for onset detections available as external objects for Pd were assessed, all of which implemented techniques based on the spectral content. Specifically, we compared the objects (i) *bonk*∼ (described in Puckette et al., 1998), which is based on the analysis of the spectral growth of 11 spectral bands; (ii) *bark*∼, from the *timbreID* library, which consists of a variation of bonk∼ relying on the Bark scale; (iii) *aubioonset*∼ from the *aubio* library, which makes available different techniques, i.e., broadband energy rise ODF (Bello et al., 2005), high frequency content ODF (Masri, 1996), complex domain ODF (Duxbury et al., 2003), phase-based ODF (Bello and Sandler, 2003), spectral difference ODF (Foote and Uchihashi, 2001), Kulback-Liebler ODF (Hainsworth and Macleod, 2003), modified Kulback-Liebler ODF (Brossier, 2006), and spectral flux-based ODF (Dixon, 2006). Several combinations of parameters were used in order to find the best performances for each method.

All these spectral methods shared in common a variable delay between the actual onset time and the time at which the onset was detected. In the end *aubioonset*∼, was selected because it was empirically found to be capable of providing the best accuracy when configured to implement the high-frequency content ODF and spectral difference ODF. Specifically, the object's arguments were: (i) for high-frequency content ODF: threshold = 0.6, buffer size = 256 samples, hop size = 64 samples; (ii) for spectral difference ODF: threshold = 3.5, buffer size = 256 samples, hop size = 64 samples. A refractory period of 30 ms was applied to both methods after such detection to eliminate false positives. The linear combination of the two spectral methods used equal weights and simply consisted in selecting the first detected onset in the time window of 30 ms.

4.1.3. Fusion Strategy

Our strategy of combining the three onset detectors calculated in parallel consisted in considering an onset if and only if both the time-based technique and the linear combination of the spectral techniques produced an onset within a time window of 20 ms. Therefore, an onset was passed to the subsequent Hit Localization and Features Extraction modules only after 20 ms from the actual initial moment of the hit. This amount of time was selected not only to accommodate the (rare) cases in which spectral methods performed with large latency, but also because the feature extraction algorithms retroactively computed their analysis on the previous 1,024 samples. Such window size corresponds to 23.2 ms (at a sampling rate of 44.1 kHz), therefore little or null pre-onset samples are included in the analysis.

4.2. Hit Detection

This module collected the onset candidates from all the five Onset Detection modules applied to each pickup, and detected the presence of a hit based on the first arriving onset. The information about the occurrence of the hit was then passed to the Features Extraction modules (applied to the signal from the condenser microphone and each of the pickups) as a trigger for their computations.

4.3. Features Extraction

This module retrieved various temporal and spectral features from the signals of the condenser microphone and pickups as soon as an onset was detected. These were used to inform the processes of hit localization, gesture classification, and sound sample selection, accomplished by the respective modules of the sound engine.

Several features and various combinations thereof were tested. Since the complexity of the feature vector increases computation time (and as a consequence latency), the goal was to choose a minimal set of features that would provide a good balance of accuracy and latency. In the following we describe the ones that proved to be the most reliable and performant features. All features were computed by using the *timbreID* library for Pd (Brent, 2010).

Notably, the features for gesture classification and sample selection were computed in parallel with the features used for hit localization in order to reduce latency.

4.3.1. Features for Hit Localization

The Hit Localization module was informed by features extracted from both the pickup signals and the condenser microphone.

Firstly, we extracted the peak of the energy from the signals of the five pickups associated to the five zones. The peak was found by searching the sample with highest energy in the 1,024 samples previous to the reporting of the onset (i.e., 20 ms after the detection of the onset). Before applying the peak picking algorithm, a threshold was applied to the energy signal to eliminate low energy components. This was followed by a linear scaling function (with clipping) to amplify or reduce the level of the signal within a range, whose extremes were empirically found. The goal of this procedure was to find a balance between the cross talking effects affecting the pickups and the amplification of the signals of the pickups closest to where the hit was produced.

From the signal of the condenser microphone we extracted three features with the goal of distinguishing sounds richer in low frequencies (BAS zone) from sounds richer in high frequencies (thus associated to the zones FTL, FTR, L, and R): the absolute value of the signal peak, the spectral brightness, and the Bark frequency spectrum. Specifically, here the spectral brightness is not intended as a perceptual measure, but as the ratio of the sum of magnitudes above a given boundary frequency to the sum of all magnitudes in a spectrum. Signals richer in high frequencies have higher brightness. We set the boundary frequency to 400 Hz as we found that signals resulting from hits on zones FTL, FTR, L, and R had more power in high frequencies above that frequency compared to those resulting from hits on BAS zone.

The Bark frequency spectrum is a warping of the normal magnitude spectrum to the Bark scale (Traunmüller, 1990). This analysis has the advantage of attenuating some of the high-frequency detail while maintaining resolution on the low end, which makes it an ideal candidate for discriminating different percussive sounds. Since the hits in the acoustic cajón contained mostly low-frequency content, only the first 10 bands were extracted. The algorithm was configured to implement a triangular filter bank spaced at half-Bark intervals.

All these calculations were performed on the window containing the 1,024 samples previous to the actual reporting of hit detection.

4.3.2. Features for Gesture Classification

Typically, the two hits classified, slap hit and tap hit, have different temporal and spectral features (e.g., attack time, peak, spectral brightness). We empirically found that the best descriptors for distinguishing such hits were the absolute value of the peak of the signal and the first 10 bands of its Bark frequency spectrum. Therefore, it was enough to reutilize the results of the same calculations performed for the Hit Localization module.

4.3.3. Features for Sample Selection and Parameters-to-Sound Mapping

Regarding the features to be passed as input for the Sample Selection and Parameters-to-Sound Mapping module, we utilized the absolute value of the peak of the signal of the condenser microphone previously computed, as well as the spectral centroid (over the previous 1,024 samples). During the training phase of the machine learning algorithm, we computed for each zone and for each gesture the minimum and maximum value of the spectral centroid. We then divided such range into four intervals to capture and render the timbral nuances associated to the hits.

4.4. Hit Localization

To localize the hit we adopted a mixed approach that involved machine learning techniques and a simple comparison of the extracted features. The process is illustrated in **Figure 4**. Firstly, we created a vector of 16 coefficients consisting of the magnitudes of the first 10 bands of the Bark frequency spectrum, the absolute value of the signal peak, the spectral brightness, 2 indices related to the two pickups with greatest peaks and their respective peak values. Secondly, we fed the classifier with such a vector and, after a training stage, we determined if the vector corresponded or not to a hit on the BAS zone. If it corresponded then the localization process ended, otherwise one of the remaining four zones was selected on the basis of the highest peak among the corresponding four pickups FTL, FTR, L, and R. Finally, the information about the localized hit was passed to the Gesture Classification module. Notably, any method based only on the comparison of the peaks of the pickup signals was found to be not sufficient to detect all zones perfectly. This was due in part to cross-talking effects and in part to the fact that a hit producing low frequencies could have as highest peak the pickups located on the FTL or FTR zones.

For the classifier we adopted the k-nearest neighbor algorithm (*k-NN*) and involved Euclidean distance measurements between feature vectors. This method was selected not only for its availability in Pd as an efficient real-time implementation, but also on the basis of the results reported in Jathal (2017), which showed that the k-NN method yielded the highest accuracy in a similar classification task (real-time timbre classification for tabletop hand drumming) compared to other methods such as support vector machine, k-means clustering, and neural networks. Specifically, we utilized the *timbreID* object from the *timbreID* library, which is capable of performing both the machine-learning function during training and the classification algorithm during testing. We set *k* (the number of neighbors to consider) to 1, as we found that it led to the best accuracy. We clustered the instances into two clusters, one for hits on zone BAS and the second for hits on zones FTL, FTR, L, and R. In particular, to boost reliability and accuracy of classification we implemented a supervised learning system by adopting a user-defined input to force manual clustering of the training data. All points composing the input vector were equally weighted.

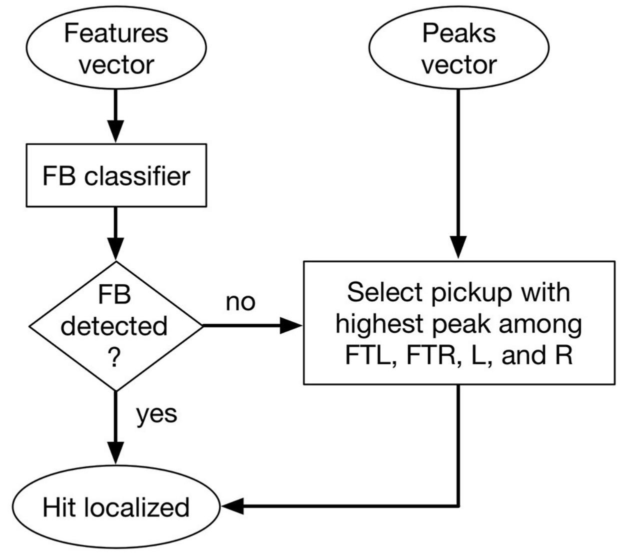

FIGURE 4 | Flow diagram of the localization process performed by the Hit Localization module.

4.5. Gesture Classification

We utilized a different classifier of the two gestures for each of the five detected zones. The five classifiers took as input an 11-point vector composed by the features described in section 4.3.2 and were configured in the same way than the classifier utilized for localization. Notably, we empirically found that extracting and classifying the features from each of the five pickups signals did not yield to better performances compared to the extraction and classification performed only on the condenser microphone's signal.

4.6. Hit Classification and Automatic Score Transcription

This module simply labeled a hit on the basis of the results of the Hit Localization and Gesture Classification modules, and passed this information to the Sample Selection and Parameters-to-Sound Mapping module. A total of 10 labels was produced (5 zones × 2 gestures): BAS_slap, BAS_tap, FTL_slap, FTL_tap, FTR_slap, FTR_tap, L_slap, L_tap, R_slap, R_tap. This module also performed an automatic transcription of the played hits in the form of a MIDI score. Each of the 10 labels were converted into a MIDI note, whose velocity was generated by mapping the absolute value of the microphone's signal peak to the range [0, 127]. Each excerpt could be saved to a MIDI file by means of record/stop commands wirelessly sent from an app for smartphone previously built (Turchet et al., 2018). The same app could be also utilized for transferring the MIDI file on a computer and to upload the file to the cloud (via FTP), thanks to a Python script running on the embedded platform.

4.7. Sample Selection, Parameters-to-Sound Mapping, and Triggering

This module took as input the 10 labels produced by the Hit Classification module, as well as the spectral centroid and absolute value of the peak as described in section 4.3.3. This information was utilized to select a sound sample among a library of 40 .wav files (5 zones × 2 gestures ×4 timbral nuances within each gesture) and to regulate its volume. The .wav files were created ad-hoc by leveraging freely available sound libraries. Specifically, we utilized sounds of a drum kit, adopting the following mapping: BAS to bass drum; FTL to snare; FTR to closed hi-hat; L to high-tom; R to crash cymbal. Finally, the selected sample was played back through the embedded loudspeaker.

5. TECHNICAL EVALUATION

We evaluated our spatio-timbral hit detection system at technical level by means of two experiments, which involved two professional cajón players (1 female, 1 male, average age = 29, average years of musical experience = 15.5). In the first experiment we aimed to assess the performance of the system using, for the involved classifiers, both training and testing data originated from the same musician. In the second experiment we tested the system performance by using a dataset of a musician against the classifiers trained with the data coming from the other musician. The goal of this second experiment was to investigate how the system is affected by the player's style and scalability.

To conduct both experiments we collected data in form of recordings, which were divided in two sessions. Recordings were performed in an acoustically isolated studio, on the same day, and using identical tuning of the instrument. In the first session, participants were instructed to play for each zone and for each gesture two sets of 50 hits. Specifically, they were asked to play such hits using different dynamics. In the second session, participants were asked to play a series of complex patterns of their choice using the five zones and the two gestures ad libitum, and involving different dynamics. These sessions were video recorded in order to annotate the hits against the zones and gestures, and assess the system performance during the testing conditions.

Recordings were analyzed offline on a Mac using the same code running on the embedded system and configuring Pd with identical settings. To assess the presence of false positives and false negatives in the detection, data were analyzed by inspecting the sound produced by the algorithms against the input signals of the microphone and of the five pickups. To assess the localization and gesture recognition performances, we analyzed the sequence of labels produced by the classification algorithms, which were saved in a log file.

5.1. Evaluation of the Onset Detector

Firstly we assessed the performance of the developed onset detector. A total of 2,348 hits was collected and annotated. Results showed that all the 2,348 hits were perfectly detected by the onset detector without any false positives. The assessment of the timing accuracy of the onset detector was conducted by calculating, over a set of 100 hits, the temporal difference between the reported hit and the actual hit. This was achieved by visually inspecting the waveform of the condenser microphone against a short burst of a square wave signal corresponding to the onset reported by the time-based technique. Results showed that an onset was detected by the time-based technique with an average accuracy of 1.72 ms and a standard error of 0.71 ms. In particular, we found that the vast majority of the times the onset was detected by means of the time-based technique before than the spectral techniques.

5.2. Results of Experiment 1

In the first experiment, the first 50 of the two collected sets of individual hits were utilized to train the classifiers. All remaining hits were utilized for testing. Results showed that all the hits were correctly localized in each of the five zones both for isolated hits and complex patterns. Success rates for the classification of the two gestures in the five zones are presented in **Table 1** for both musicians. The average success rate considering all the gestures was 96.16%. Results in terms of precision, recall, and F-measure statistics are reported in **Table 2**.

5.3. Results of Experiment 2

In the second experiment, we utilized the database created with the training data of the first musician to test the performance of the system when using all the hits collected from the second musician, and vice versa. The utilized training data were the same ones involved in the first experiment. The set of testing data from the first performer amounted to 1,213 hits, that of the second performer to 1,135 hits. Results showed that all hits of the first performer were localized correctly using the dataset of the second performer. Two erroneous identifications were reported for the hits of the second performer, where the FTL zone was identified as BAS zone. Gesture identification performances in terms of precision, recall, and F-measure statistics are reported in **Table 3** for each zone and for both musicians.

6. DISCUSSION AND CONCLUSIONS

The results of the first experiment showed that hit detection and localization could be always classified correctly, and gesture identification had generally high success rates for the two gestures in all the five zones when data from a single player were considered. Gesture identification was at comparable level with that of the study reported in Jathal (2017) for a tabletop drum, which involved a similar timbre-recognition system.

These results were achieved thanks to a dedicated hardware architecture composed by a system of pickups and a condenser microphone, as well as a sound engine. On the one hand, the engine leveraged a novel onset detector that was based on the combination of a time-based method (with minimal latency but more prone to false positives) and two spectral methods (more robust to false positives). On the other hand, the engine utilized a robust features extraction system capturing the temporally evolving timbral and temporal characteristics in the first 20 ms of a hit sound, as well as a feature-based K-nearest neighbor

TABLE 1 | Success rates in Experiment 1 for the identification of the slap and tap gestures for each zone.

Hit	Performer 1		Performer 2		% correct
	No. hits	No. correct	No. hits	No. correct	
BAS_Slap	69	69	63	63	100
BAS_Tap	51	45	73	72	94.35
FTL_Slap	86	84	80	78	97.59
FTL_Tap	94	92	52	52	98.63
FTR_Slap	84	80	69	69	97.38
FTR_Tap	84	84	64	64	100
L_Slap	54	53	59	49	90.26
L_Tap	56	54	51	45	92.52
R_Slap	59	58	68	60	92.91
R_Tap	51	51	52	50	98.05

TABLE 2 | System performances during Experiment 1 for the classification of both gestures in each zone according to precision (p), recall (r), and F-measure (F) statistics.

Hit	Performer 1			Performer 2			Total (mean ± standard error)		
	p	*r*	*F*	*p*	*r*	*F*	*p*	*r*	*F*
BAS_Slap	0.92	1	0.95	0.98	1	0.99	0.95 ± 0.03	1 ± 0	0.97 ± 0.01
BAS_Tap	1	0.88	0.93	1	0.98	0.99	1 ± 0	0.93 ± 0.05	0.96 ± 0.02
FTL_Slap	0.97	0.97	0.97	1	0.97	0.98	0.98 ± 0.01	0.97 ± 0	0.98 ± 0
FTL_Tap	0.97	0.97	0.97	0.96	1	0.98	0.97 ± 0	0.98 ± 0.01	0.97 ± 0
FTR_Slap	1	0.95	0.97	1	1	1	1 ± 0	0.97 ± 0.02	0.98 ± 0.01
FTR_Tap	0.95	1	0.97	1	1	1	0.97 ± 0.02	1 ± 0	0.98 ± 0.01
L_Slap	0.96	0.98	0.97	0.89	0.83	0.85	0.92 ± 0.03	0.9 ± 0.07	0.91 ± 0.05
L_Tap	0.98	0.96	0.97	0.81	0.88	0.84	0.9 ± 0.08	0.92 ± 0.04	0.91 ± 0.06
R_Slap	1	0.98	0.99	0.96	0.88	0.92	0.98 ± 0.01	0.93 ± 0.05	0.95 ± 0.03
R_Tap	0.98	1	0.99	0.86	0.96	0.9	0.92 ± 0.05	0.98 ± 0.01	0.94 ± 0.04

TABLE 3 | System performances during Experiment 2 for the classification of both gestures in each zone according to precision (p), recall (r), and F-measure (F) statistics.

	Performer1			Performer2			Total (mean ± standard error)		
Hit	*p*	*r*	*F*	*p*	*r*	*F*	*p*	*r*	*F*
BAS_Slap	1	0.34	0.51	0.49	1	0.66	0.74 ± 0.25	0.67 ± 0.32	0.58 ± 0.07
BAS_Tap	0.56	1	0.72	1	0.07	0.13	0.78 ± 0.21	0.53 ± 0.46	0.42 ± 0.29
FTL_Slap	0.41	0.61	0.49	0.56	1	0.72	0.48 ± 0.07	0.8 ± 0.19	0.6 ± 0.11
FTL_Tap	0.1	0.04	0.06	1	0	0.01	0.55 ± 0.44	0.02 ± 0.01	0.04 ± 0.02
FTR_Slap	0.89	0.5	0.64	0.47	0.87	0.61	0.68 ± 0.2	0.69 ± 0.18	0.63 ± 0.01
FTR_Tap	0.65	0.94	0.77	0.06	0	0.01	0.35 ± 0.29	0.47 ± 0.46	0.39 ± 0.37
L_Slap	0.44	0.77	0.56	0.52	1	0.68	0.48 ± 0.03	0.88 ± 0.11	0.62 ± 0.05
L_Tap	0.17	0.04	0.07	1	0	0.01	0.58 ± 0.41	0.02 ± 0.01	0.04 ± 0.02
R_Slap	0.63	0.65	0.64	0.54	0.97	0.69	0.59 ± 0.04	0.81 ± 0.16	0.67 ± 0.02
R_Tap	0.61	0.6	0.61	0.66	0.05	0.1	0.64 ± 0.02	0.33 ± 0.27	0.35 ± 0.25

classifier. Therefore, our approach adopted a combination of both sensor augmentation and indirect acquisition, which are the two main techniques currently utilized to sense gestures performed on musical instruments (Driessen and Tzanetakis, 2018). Specifically, such a sensor fusion was conceived to cope with the limitations of systems based on the sole information coming from the pickups, since this was found to be not enough to detect in a satisfactory way the variety of techniques that can be used by different musicians. Indeed, the participants in our study exhibited rather different playing styles and the system was capable of adapting to both of them.

Such differences between players were investigated in the second experiment to assess the performances of the system using the classifiers trained with the data of one performer and tested with the data of the other performer (and vice versa). On the one hand, results showed that the binary classifier used to distinguish the BAS zone from the other zones performed almost perfectly. On the other hand, the precision, recall, and F-measure scores for gestures identification drastically decreased compared to the first experiment. This finding suggests that hit localization might be scaled relatively safely across musicians, but gesture identification seems requiring a training specific for each musician.

There are a few areas of technical optimization that could be further explored. Firstly, a sequential k-means clustering technique (Hartigan and Wong, 1979) could be used in place of the k-NN technique, following a similar approach than that reported in Miron et al. (2013a).

Secondly, the involved classification algorithm could be informed by a larger dataset during the training process to improve accuracy. However, this might increase not only the computational load, but also latency since the real-time performance of a k-NN-based classifier depends on the number of trained points. Therefore, the challenge will be to find, via empirical experimentation, a tradeoff between improvements in reliability of classification due to a large dataset and the resulting increase in latency to the extent that can be perceptually tolerated by cajón players, at the same time considering the available computational efficiency provided by the utilized hardware architecture.

Thirdly, the feature set was chosen through empirical experimentation and from spectral analyses among a larger set of features, where not all possible combinations where explored. Further enhancement to our system might be provided by statistical procedures involving heuristics and genetic algorithms to achieve a deeper timbral analysis (Witten et al., 2016).

In order to recreate ecologically-valid conditions, the system was tested in presence of various concurrent music pieces provided by two loudspeakers at relatively loud volumes. This proved that typical external musical sounds do not interfere with the system. As a matter of fact, the utilized condenser microphone and the miking technique consisting of placing it inside the instrument are the same utilized for amplifying conventional acoustic cajones during professional performances. On the other hand, thresholds on the contact microphones signals were tuned to respond exclusively to actual hits of the player, discarding any other type of vibration.

It has to be noticed that this study suffers from the limitation that only one exemplar of acoustic cajón was involved. The timbre of acoustic cajones may vary drastically, and therefore the proposed technique may need to be revised or their parameters tuned differently. Effective unsupervised learning method should be devised and implemented for this purpose.

Another limitation of the proposed approach is its inability of tracking compound hits, which might happen. We plan to face this challenge in future works. Techniques such as instance filtering proposed in Miron et al. (2013a) seem promising for this purpose. This technique is based on a stage where overlapping events, such as simultaneous hits on different components of a drum kit, are filtered before being passed to the onset detection, feature extraction, and classification processes (using k-NN or k-means). We also plan to track more gestures such as those involving nails or knuckles, which are typically less common.

Real-time hit detection of percussive sounds opens new possibilities for research, such as the detection of patterns involving the different locations. This would require the use of pattern recognition methods adapted for real-time contexts (Goto, 2001). The resulting information could be exploited or expressive purposes, such as real-time control of digital audio effects (Holfelt et al., 2017) or of external equipment by exploiting the wireless communication capabilities of the smart caón (see e.g., Turchet and Barthet, 2017).

More importantly, in future work we plan to perform a perceptual validation involving professional musicians as a further iteration in our user-centered design approach (Turchet et al., 2018).

ETHICS STATEMENT

This study was carried out in accordance with the recommendations of the Ethical Committee of Queen Mary University of London. The protocol was approved by the Ethical Committee of Queen Mary University of London. All subjects gave written informed consent in accordance with the Declaration of Helsinki.

AUTHOR CONTRIBUTIONS

LT: conception, design of the work, data acquisition, data analysis, data interpretation, drafting the work, revising the work, final approval, accountable for all aspects of the work; AM and MB: revising the work, final approval, accountable for all aspects of the work.

FUNDING

The work of LT is supported by a Marie-Curie Individual fellowship from the European Union's Horizon 2020 research and innovation programme (749561) as well as by a Micro Grant of the Marie Curie Alumni Association. AM is supported by a grant of the Engineering and Physical Sciences Research Council (EP/N005112/1). MB acknowledges support from the EU H2020 Audio Commons grant (688382).

REFERENCES

Aimi, R. M. (2007). *Hybrid Percussion: Extending Physical Instruments Using Sampled Acoustics*. Ph.D. thesis, Massachusetts Institute of Technology.

Barthet, M., Depalle, P., Kronland-Martinet, R., and Ystad, S. (2010). Acoustical correlates of timbre and expressiveness in clarinet performance. *Music Percept.* 28, 135–154. doi: 10.1525/mp.2010.28.2.135

Bello, J., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. (2005). A tutorial on onset detection in music signals. *IEEE Trans. Speech Audio Process.* 13, 1035–1047. doi: 10.1109/TSA.2005.851998

Bello, J., and Sandler, M. (2003). "Phase-based note onset detection for music signals," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing* (Hong Kong). 441–444.

Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., and Klapuri, A. (2013). Automatic music transcription: challenges and future directions. *J. Intell. Inform. Syst.* 41, 407–434. doi: 10.1007/s10844-013-0258-3

Brent, W. (2009). "Cepstral analysis tools for percussive timbre identification," in *Proceedings of the International Pure Data Convention* (São Paulo).

Brent, W. (2010). "A timbre analysis and classification toolkit for pure data," in *Proceedings of the International Computer Music Conference* (New York, NY).

Brossier, P. (2006). *Automatic Annotation of Musical Audio for Interactive Systems*. Ph.D. thesis, Queen Mary University of London.

Brossier, P., Bello, J., and Plumbley, M. (2004). "Real-time temporal segmentation of note objects in music signals," in *Proceedings of the International Computer Music Conference* (Miami, FL).

Burgoyne, J., Fujinaga, I., and Downie, J. (2016). "Music information retrieval," in *A New Companion to Digital Humanities*, eds S. Schreibman, R. Siemens, and J. Unsworth (Hoboken, NJ: John Wiley & Sons, Ltd.), 213–228. doi: 10.1002/9781118680605

Dixon, S. (2006). "Onset detection revisited," in *Proceedings of the International Conference on Digital Audio Effects* (Montréal, QC), 133–137.

Driessen, P., and Tzanetakis, G. (2018). "Digital sensing of musical instruments," in *Springer Handbook of Systematic Musicology*, ed R. Bader (Berlin; Heidelberg: Springer), 923–933.

Duxbury, C., Bello, J., Davies, M., and Sandler, M. (2003). "Complex domain onset detection for musical signals," in *Proceedings of the Digital Audio Effects Conference* (London), 1–4.

Finney, S. A. (1997). Auditory feedback and musical keyboard performance. *Music Percept.* 15, 153–174.

Foote, J., and Uchihashi, S. (2001). "The beat spectrum: a new approach to rhythm analysis," in *Proceedings of IEEE International Conference on Multimedia and Expo* (Tokyo), 881–884.

Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drum-sounds. *J. New Music Res.* 30, 159–171. doi: 10.1076/jnmr.30.2.159.7114

Gouyon, F., and Herrera, P. (2001). "Exploration of techniques for automatic labeling of audio drum tracks instruments," in *Proceedings of MOSART: Workshop on Current Directions in Computer Music*.

Hainsworth, S., and Macleod, M. (2003). "Onset detection in musical audio signals," in *Proceedings of the International Computer Music Conference* (Singapore).

Hartigan, J. A., and Wong, M. A. (1979). Algorithm as 136: a k-means clustering algorithm. *J. R. Stat. Soc. Ser. C Appl. Stat.* 28, 100–108.

Herzog, S. (2013). "Efficient DSP implementation of median filtering for real-time audio noise reduction," in *Proceedings of the International Conference on Digital Audio Effects* (Maynooth), 1–6.

Holfelt, J., Csapo, G., Andersson, N., Zabetian, S., Castenieto, M., Dahl, S., et al. (2017). "Extraction, mapping, and evaluation of expressive acoustic features for adaptive digital audio effects," in *Proceedings of the Sound & Music Computing Conference* (Espoo).

Jathal, K. (2017). Real-time timbre classification for tabletop hand drumming. *Comput. Music J.* 41, 38–51. doi: 10.1162/COMJ_a_00419

Jensenius, A. R., and Wanderley, M. M. (2010). "Musical gestures: concepts and methods in research," in *Musical Gestures*, eds R. I. Godøy and M. Leman (London; New York, NY: Routledge), 24–47.

Kapur, A., Davidson, P., Cook, P., Driessen, P., and Schloss, W. (2004). "Digitizing North Indian Performance," in *Proceedings of the International Computer Music Conference* (Miami, FL).

Liggins, M., Hall, D., and Llinas, J. (eds.). (2017). *Handbook Multisensor Data Fusion: Theory and Practice, 2nd Edn*. Boca Raton, FL: CRC Press.

Ludwigsen, D. (2017). "Acoustic and structural resonances of the cajon," in *Proceedings of Meetings on Acoustics 170 ASA* (Jacksonville, FL).

MacRitchie, J., and McPherson, A. P. (2015). Integrating optical finger motion tracking with surface touch events. *Front. Psychol.* 6:702. doi: 10.3389/fpsyg.2015.00702

Masri, P. (1996). *Computer Modelling of Sound for Transformation and Synthesis of Musical Signals*. Ph.D. thesis, University of Bristol, Department of Electrical and Electronic Engineering.

McPherson, A., and Zappi, V. (2015). "An environment for Submillisecond-Latency audio and sensor processing on BeagleBone black," in *Audio Engineering Society Convention 138* (New York, NY).

McPherson, A. P., Jack, R. H., and Moro, G. (2016). "Action-sound latency: are our tools fast enough?," in *Proceedings of the Conference on New Interfaces for Musical Expression* (Brisbane, QLD).

Michalakos, C. (2012). "The augmented drum kit: an intuitive approach to live electronic percussion performance," in *Proceedings of the International Computer Music Conference* (Ljubljana).

Miranda, E., and Wanderley, M. (2006). *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*, *Vol. 21*. Middleton, WI: AR Editions, Inc.

Miron, M., Davies, M., and Gouyon, F. (2013a). "Improving the real-time performance of a causal audio drum transcription system," in *Proceedings of the Sound and Music Computing Conference* (Stockholm), 402–407.

Miron, M., Davies, M., and Gouyon, F. (2013b). "An open-source drum transcription system for pure data and MAX MSP," in *IEEE International Conference on Acoustics, Speech and Signal Processing* (Vancouver, BC), 221–225.

Mitchell, T., Madgwick, S., Rankine, S., Hilton, G., Freed, A., and Nix, A. (2014). "Making the most of wi-fi: Optimisations for robust wireless live music performance," in *Proceedings of the Conference on New Interfaces for Musical Expression* (London), 251–256.

Moore, B. (2012). *An Introduction to the Psychology of Hearing*. Bingley: Brill.

Moore, F. R. (1988). The dysfunctions of midi. *Comput. Music J.* 12, 19–28.

Odowichuk, G., Trail, S., Driessen, P., Nie, W., and Page, W. (2011). "Sensor fusion: towards a fully expressive 3d music control interface," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* (Victoria, BC), 836–841.

Pardue, L., Harte, C., and McPherson, A. (2015). A low-cost real-time tracking system for violin. *J. New Music Res.* 44, 305–323. doi: 10.1080/09298215.2015.1087575

Puckette, M., Apel, T., and Ziccarelli, D. (1998). "Real-time audio analysis tools for PD and MSP," in *Proceedings of the International Computer Music Conference* (Ann Arbor, MI).

Stowell, D., and Plumbley, M. (2007). "Adaptive whitening for improved real-time audio onset detection," in *Proceedings of the International Computer Music Conference* (Copenhagen), 312–319.

Tian, M., Fazekas, G., Black, D., and Sandler, M. (2014). "Design and evaluation of onset detectors using different fusion policies," in *Proceedings of International Society for Music Information Retrieval Conference* (Taipei), 631–636.

Tindale, A., Kapur, A., and Tzanetakis, G. (2011). Training surrogate sensors in musical gesture acquisition systems. *IEEE Trans. Multimedia* 13, 50–59. doi: 10.1109/TMM.2010.2089786

Tindale, A. R., Kapur, A., Schloss, W. A., and Tzanetakis, G. (2005). "Indirect acquisition of percussion gestures using timbre recognition," in *Proceedings of the Conference on Interdisciplinary Musicology* (Montréal, QC).

Tindale, A. R., Kapur, A., Tzanetakis, G., and Fujinaga, I. (2004). "Retrieval of percussion gestures using timbre classification techniques," in *Proceedings of International Society for Music Information Retrieval Conference* (Barcelona).

Tompkins, W. D. (2007). "Afro-peruvian traditions," in *The Garland Handbook of Latin American Music*, eds D. A. Olsen and D. E. Sheeh (London; New York, NY: Routledge), 474–487.

Traunmüller, H. (1990). Analytical expressions for the tonotopic sensory scale. *J. Acoust. Soc. Am.* 88, 97–100.

Turchet, L. (2018). "Hard real time onset detection for percussive sounds," in *Proceedings of the Digital Audio Effects Conference* (Aveiro).

Turchet, L., and Barthet, M. (2017). "Envisioning smart musical haptic wearables to enhance performers' creative communication," in *Proceedings of International Symposium on Computer Music Multidisciplinary Research* (Porto), 538–549.

Turchet, L., Fischione, C., and Barthet, M. (2017). "Towards the internet of musical things," in *Proceedings of the Sound and Music Computing Conference* (Espoo), 13–20.

Turchet, L., McPherson, A., and Barthet, M. (2018). Co-design of a smart Cajón. *J. Audio Eng. Soc.* 66, 220–230. doi: 10.17743/jaes.2018.0007

Turchet, L., McPherson, A., and Fischione, C. (2016). "Smart instruments: towards an ecosystem of interoperable devices connecting performers and audiences," in *Proceedings of the Sound and Music Computing Conference* (Hamburg), 498–505.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Burlington: Morgan Kaufmann.

Young, D., and Fujinaga, I. (2004). "Aobachi: a new interface for japanese drumming," in *Proceedings of the Conference on New Interfaces for Musical Expression* (Hamamatsu), 23–26.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.