Training Surrogate Sensors in Musical Gesture Acquisition Systems

Adam Tindale, Ajay Kapur, and George Tzanetakis, *Member, IEEE*

***Abstract*—Capturing the gestures of music performers is a common task in interactive electroacoustic music. The captured gestures can be mapped to sounds, synthesis algorithms, visuals, etc., or used for music transcription. Two of the most common approaches for acquiring musical gestures are: 1) "hyper-instruments" which are "traditional" musical instruments enhanced with sensors for directly detecting the gestures and 2) "indirect acquisition" in which the only sensor is a microphone capturing the audio signal. Hyper-instruments require invasive modification of existing instruments which is frequently undesirable. However, they provide relatively straightforward and reliable sensor measurements. On the other hand, indirect acquisition approaches typically require sophisticated signal processing and possibly machine learning algorithms in order to extract the relevant information from the audio signal. The idea of using direct sensor(s) to train a machine learning model for indirect acquisition is proposed in this paper. The resulting trained "surrogate" sensor can then be used in place of the original direct invasive sensor(s) that were used for training. That way, the instrument can be used unmodified in performance while still providing the gesture information that a hyper-instrument would provide. In addition, using this approach, large amounts of training data can be collected with minimum effort. Experimental results supporting this idea are provided in two detection contexts: 1) strike position on a drum surface and 2) strum direction on a sitar.**

***Index Terms*—Gesture recognition, machine learning, new interfaces for musical expression, surrogate sensors, virtual sensors.**

I. Introduction

THROUGHOUT history, musical instruments have been some of the best examples of artifacts designed for interaction. In recent years, a combination of cheaper sensors, more powerful computers, and rapid prototyping software has resulted in a plethora of interactive electroacoustic music performances and installations. In many of these performances, traditional acoustic instruments are blended with computer-generated sounds and visuals. Automatically sensing the gestures made by the performer is frequently desired in such interactive multimedia performances. For example, we might be interested in the strumming pattern of a guitar player or we might be interested how hard a pianist strikes a chord. This extracted information has been used in several ways including driving interactive graphics syncrhonized to the music, having computers or robots react to the music performed, and to gain a more detailed quantitative aspects of music performance such as the nuances of timing.

The extraction of information from musical instruments provides a fascinating domain to explore ideas of multimedia processing beyond the more traditional audio, image, and video processing that is the currently the dominant focus of multimedia research. A combination of different sensors can be utilized, and typically their output needs to be further processed by a combination of digital signal processing and machine learning techniques to extract useful information. A further challenge is that the information needs to be extracted causally and in real-time in order to be utilized in live music performance. Therefore, an interactive computer-music performance is a great example of a multimodel human-computer interface in action. The work presented in this paper grew out of the experiences of the authors in developing instruments for live interactive human-computer music performances.

There are two main approaches to sensing instrumental gestures. In indirect acquisition, traditional acoustical instruments are extended/modified with a variety of sensors such as force sensing resistors (FSR), and accelerometers. The purpose of these sensors is to measure various aspects of the gestures of the performers interacting with their instruments. A variety of such "hyper-instruments" have been proposed [1]–[3]. However, there are many pitfalls in creating such sensor-based controller systems. Purchasing microcontrollers and certain sensors can be expensive. The massive tangle of wires interconnecting one unit to the next can get failure-prone. Things that can go wrong include: simple analog circuitry break down, or sensors wearing out right before a performance forcing musicians to carry a soldering iron along with their tuning fork. However, the biggest problem with hyper-instruments is that there usually is only one version. Therefore, only one performer, typically the designer/builder, can benefit from the data acquired and utilize the instrument in performances. Finally, musical instruments, especially the ones played by professionals, can be very expensive, and therefore, any invasive modification to attach sensors is bound to be met with resistance if not absolute horror.

These problems have motivated researchers to work on indirect acquisition in which the musical instrument is not modified in any way. The only input is provided by non-invasive sensors, typically one or more microphones. The recorded audio then needs to be analyzed in order to measure the various desired gestures. Probably the most common and familiar example of indirect acquisition is the use of automatic pitch detectors to turn monophonic acoustic instruments into music instrument

Manuscript received March 22, 2010; revised July 20, 2010; accepted October 08, 2010. Date of publication October 28, 2010; date of current version January 19, 2011. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nicu Sebe.

The authors are with the Department of Computer Science, the Department of Electrical Engineering, and the Faculty of Music, University of Victoria, Victoria, BC V8S 1P2, Canada (e-mail: art@uvic.ca; akapur@alumni.princeton.edu; gtzan@cs.uvic.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2010.2089786

digital interface (MIDI) instruments. In most cases, indirect acquisition does not directly capture the intended measurement and the signal needs to be analyzed further to extract the desired information. Frequently this analysis is achieved by using real-time signal processing techniques. More recently, an additional stage of supervised machine learning has been utilized in order to "train" the information extraction algorithm. The disadvantage of indirect acquisition is the significant effort required to develop the signal processing algorithms. In addition, if machine learning is utilized, the training of the system can be time consuming and labor intensive.

The main problem addressed in this paper is the efficient and effective construction of indirect acquisition systems for musical instruments in the context of interactive media. Our proposed solution is based on the idea of using direct sensors to train machine learning models that predict the direct sensor outputs from acoustical data. Once these indirect models have been trained and evaluated, they can be used as "surrogate" sensors in place of the direct sensors. This approach is motivated by ideas in multimodal data fusion with the slight twist that in our case, the data fusion is only used during the learning phase. We believe that the idea of using direct sensors to learn mappings for indirect acquisition can be applied to other area of multimodal interaction in addition to musical instruments.

This approach of using direct sensors to "learn" indirect acquisition models has some nice characteristics. Large amounts of training data can be collected with minimum effort just by playing the enhanced instrument with the sensors. Once the system is trained and provided the accuracy and performance of the learned surrogate sensor is satisfactory, there is no need for direct sensors or invasive modifications to the instrument.

The traditional use of machine learning in audio analysis has been in classification where the output of the system is an ordinal value (for example, the instrument name). As a first case study of our proposed method, we describe a system for classifying percussive gestures using indirect acquisition. More specifically, the strike position of a stick on a snare drum is automatically inferred from the audio recording. A radio drum controller is used as the direct sensor in order to train the indirect acquisition. In addition, we explore regression which refers to machine learning systems where the output is a continuous variable. One of the challenges in regression is obtaining large amounts of data for training which is much easier using our proposed approach. In our experiments, we use audio-based feature extraction with synchronized continuous sensor data to train a "surrogate" sensor using machine learning. More specifically, we describe experiments using the electronic sitar (E-Sitar), a digitally enhanced sensor-based controller modeled after the traditional North Indian sitar. The case studies were motivated by the specific needs and knowledge of the authors during the creation of interactive computer music performances. As our goal has been in addition to research to use these techniques successfully in live music performance, it is important to involve trained musicians (which all of the authors are) that have extensive experience with playing a particular instrument. For example, the sensor extraction on the E-Sitar has been used in performance of a sitar player interacting with a robotic percussionist that is able to vary the rhythmic accompaniment and follow the expressive timing of the sitar performer. The drum strike location has been used in live music performance for changing the parameters of synthesized percussive sound in a continuous manner. These are only some of the possibilities afforded by better sensing in the context of interactive computer music performance.

We believe that the more general idea of "surrogate" sensor training can be applied to other music instruments and multimedia contexts and discuss some possibilities in the last section.

II. Background

The use of sensors to gather gestural data from a musician has been used as an aid in the creation of real-time computer music performance. In the last few years the New Interfaces for Musical Expression (NIME) conference has been the main forum for advances in that area. Some representative examples of such systems are: the Hypercello [1], the digitized Japanese drum Aobachi [3], and the E-Sitar [2]. All these hyper-instruments still function as acoustical instruments but are enhanced with a variety of direct sensors to capture gestures of the performer. Examples of information measured by the sensors include: bowing pressure and speed, strike force, and fret location. That information has been used to drive interactive graphics and sound, change the parameters of sound synthesis algorithms [4], and coordinate the human performer with computer generated sounds and accompaniment in some cases including computer control music robots [5], [6]. Another interesting application is the quantitative analysis of music performance. A general overview of new digital musical instruments including hyper-instruments can be found in Miranda and Wandeley [7].

In addition, there has been some research using machine learning techniques [8] to classify specific gestures based on audio feature analysis. The extraction of control features from the timbre space of the clarinet is explored in [9]. Deriving gesture data from acoustic analysis of a guitar performance is explored in [10]–[12]. An important influence for our research is the concept of indirect acquisition of instrumental gesture described in [12]. In that work, the audio signal generated from a classical guitar is processed using signal processing to extract which string of the guitar is played when a particular note is sounded (the same note can be played on different strings in the guitar with subtle but noticeable differences in timbre). Gesture extraction from drums is explored in [13]–[15]. The proposed algorithms rely on signal processing possibly followed by machine learning to extract information. Typically the information is categorical in nature, for example, the type of drum sound played (for example, snare, bass drum, or cymbal). In such approaches, a large number of drum sounds are collected, labeled manually, and then used with audio feature extraction to train machine learning models.

In this paper, we address the challenge of collecting large amounts of training data without needing to manually label recordings. Direct sensors are used to automatically annotate the recordings. Once the indirect acquisition method has achieved satisfactory performance, the direct sensors can be discarded. Collecting large amounts of data becomes simply playing the instrument. Most existing indirect acquisition methods make categorical decisions (classification). Using

regression [16], it is possible to deal with continuous gestural data in a machine learning framework. However, training regression models requires more data which is much easier using the proposed approach rather than manual labeling.

The concept of "virtual" sensors typically refers to the creation of software-based sensors that combine readings from several potentially heterogeneous sources to a single measurement [17]. A simple example would be a position sensor that uses GPS but switches to more accurate local position sensors when inside a particular building. The "virtual" sensor essentially abstracts this process into a single position measurement. Frequently the programmer needs to explicitly define the mapping of the "physical" sensors to the "virtual" sensors.

More recently, machine learning techniques have been used for a variety of sensor-related tasks for which direct modeling can either not be used or is difficult to formulate. Artificial neural networks are a technique frequently utilized for classification problems [18], [19] but other approaches such as support vector machines have also been used [20]. An interesting extension to using machine learning in sensor applications is creating "virtual" sensors by utilizing trained "black-box" models to perform the mapping rather than explicit programming [21]. This is particularly valuable when the underlying physics are too complex to model while there is plenty of data to develop/train a "virtual" sensor. Such sensors have many uses in automotive applications [19].

We use the term "surrogate sensor" to refer to the process of using a "physical" sensor to train a machine learning model for a "virtual sensor". For example, in automotive applications, laboratory-quality expensive sensors can be used to provide ground-truth for training a "virtual" sensor that takes input from several low-grade production-quality on-board sensors [22]. In this paper, we describe how surrogate sensors can be applied in the context of acquiring performance information using sensors on musical instruments. The advantage of using the technology in a musical context is that the cost of failure is very low compared to automotive applications: a missed note has much less impact on the user than a failure of a crash sensor. Surrogate sensors do not require any modification to the instrument as they operate only on features calculated from the audio signal captured by a microphone. Using this approach significantly simplifies the training process as it does not require any manual labeling and large amounts of annotated training data can be simply be collected by playing the instrument. In addition, it facilitates adoption by musicians as it does not require any modification to their musical instrument.

This paper expands on earlier work by the authors in the context of sitar [23] and drum performance [24] by providing a more complete description of the process of integrating sensors, digital signal processing, and machine learning using the idea of "surrogate" sensors. Additional experimental results that include classification, ordinal regression, and regression tasks are also reported.

III. System Overview

Fig. 1 shows a schematic diagram of the training process for surrogate sensors. The process has two phases: training and performance. In training, the musician plays a instrument that has been modified with additional direct physical sensors. In addition, a microphone is used to capture the audio generated by the instrument. The audio signal is analyzed using digital signal processing techniques and a compact feature representation is automatically extracted. The physical sensor readings are time-aligned with the stream of feature vectors and used as ground-truth to train machine learning models for mapping the feature vectors to the desired sensor measurement. Large amounts of training data can be collected this way as there is no need for any manual input other than the performer playing the instrument. This is in contrast to traditional approaches that rely on manual annotation of the audio signal after acquisition for creating the ground-truth labeling.

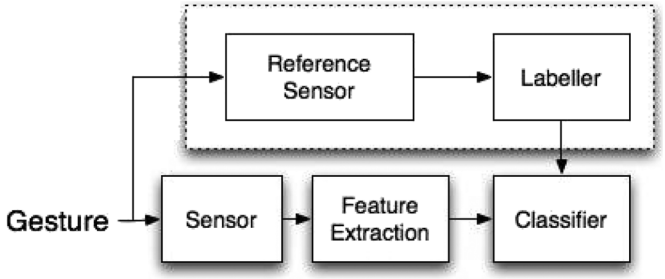

Fig. 1. nSystem diagram of surrogate sensor lifting. Once training is complete, the blocks in dotted lines are eliminated.

Once the machine learning model achieves satisfactory performance, it can be stored and used for the creation of a surrogate sensor. The surrogate sensor will behave similarly to the original invasive physical sensor but will operate on the features extracted from audio. After training, the invasive physical sensors can be removed and the performer can play an unmodified instrument while still capturing performance information using the surrogate sensor instead of the physical sensor.

It is important to briefly comment on the generalization of the surrogate sensor to other contexts. In the most restrictive context, the sensor is used on the exact same instrument and by the same performer. For the gestures explored in this paper, we have found that the trained surrogate sensor typically generalizes well to other performers playing the same instrument. In terms of generalizing to different particular instruments of the same type, it depends on the particulars. For example, trained surrogate sensors generalize well to snare drums that are of the same type as the one used for training. When the sound of the instrument is significantly different, even if it is the same instrument, the surrogate sensor does not generalize as well. Another issue that needs to be briefly discussed is the use of the "surrogate" sensor in music performance where there is a complex mixture of sounds present. In our performances, we utilize standard directional microphones that are either close to the instrument being played or part of it. Although there is some leakage of ambient noise, it does not seem to have an effect on the performance of the audio analysis. Such microphones are almost always already present in the context of music performance for recording purposes.

The remainder of the paper is structured as follows: Section IV describes the specific details of the experimental setup used for experiments with gesture acquisition for two

Fig. 2. E-Sitar and thumb sensor.

music instruments: sitar, a North Indian string instrument, and a regular snare drum. In addition, the audio feature extraction and learning process used in the experiments is described. Section V describes the experimental results for these two case studies and Section VI concludes the paper and describes directions for future work.

IV. Measurement System Configuration

A. E-Sitar

The sitar is a 19-stringed, pumpkin shelled, traditional North Indian instrument. Its bulbous gourd (shown in Fig. 2), cut flat on the top, is joined to a long necked hollowed concave stem that stretches three feet long and three inches wide. The sitar contains seven strings on the upper bridge, and twelve sympathetic strings below. All strings can be tuned using tuning pegs. The upper strings include rhythm and drone strings, known as chikari. Melodies, which are primarily performed on the uppermost string and occasionally the second copper string, induce sympathetic resonances in the twelve strings below. The sitar can have up to 22 moveable frets, tuned to the notes of a Raga (the melodic mode, scale, order, and rules of a particular piece of Indian classical music) [25].

It is important to understand the traditional playing style of the sitar to comprehend how our controller captures its hand gestures. Our controller design has been informed by the needs and constraints of the long tradition and practice of sitar playing. The sitar player uses his left index finger and middle finger, as shown in Fig. 3, to press the string to the fret to play the desired swara (note). The frets are elliptically curved so the string can be pulled downward, to bend to a higher note. This is how a performer incorporates the use of shruti (microtones) which is an essential characteristic of traditional classical Indian music. On the right index finger, a sitar player wears a ring like plectrum, known as a *mizrab*. The right hand thumb, remains securely on the edge of the dand (neck) as shown in Fig. 3, as the entire right hand gets pulled up and down over the main seven strings, letting the *mizrab* strum the desired melody. An upward stroke is known as Dha and a downward stroke is known as Ra [25]. The two main gestures we capture using sensors and subsequently try to model using audio-based analysis are: 1) the pitch/fret position and 2) the *mizrab* stroke direction.

Fig. 3. E-Sitar and thumb sensor.

The E-Sitar was built with the goal of capturing a variety of gestural input data. A more detailed description of audio-based gesture extraction on the E-Sitar including monophonic pitch detection can be found in [16]. A variety of different sensors such as fret detection using a network of resistors are used combined with an Atmel AVR ATMega16 microcontroller for data acquisition. Fig. 4 shows a schematic diagram of the resistor network used to detect the fret played. The fret detection operates by a network of resistors attached in series to each fret on the E-Sitar. Voltage is sent through the string, which establishes a connection when the string is pressed down to a fret. This results in a unique voltage based on the amount of resistance in series up to that fret. The voltage is then calculated and transmitted using the MIDI protocol.

The direct sensor used to deduce the direction of a *mizrab* stroke is a force sensing resistor (FSR), which is placed directly under the right hand thumb, as shown in Fig. 2. The thumb never moves from this position while playing; however, the applied force varies based on the *mizrab* stroke direction. A *Dha* stroke (upward stroke) produces more pressure on the thumb than a *Ra* stroke (downward stroke). We send a continuous stream of data from the FSR via MIDI, because this data is rhythmically in time and can be used compositionally for more than just deducing pluck direction. A vector of audio features is extracted and the values of the FSR sensor are fused and used to train the surrogate sensor using a regression model. More details about the experiments are provided below.

B. E-Snare

For this project, the position of the drum strike is the primary gesture for recognition. With an acoustic drum, the timbre changes as the strike moves from the center of the drum to the edge. Drummers can utilize this change in timbre when playing to create different sound textures. Very few electronic percussion devices include this feature, and thus lower the expressive potential for drummers. Strike position is measured as the distance from the center to the edge of the drum surface. Two different drum surfaces were employed for this process: an acoustic snare drum and an electronic drum pad.

The acoustic snare drum is a standard drumset component that has a 14-inch diameter and metal wires (snares) attached to the underside that vibrate against the drum. The snares may be disengaged to produce a more traditional drum sound. The acoustic snare drum was recorded using a Shure SM-57 microphone placed at the edge of the drum.

Electronic drum pads are components of electronic drumsets. The pad used was had a diameter of 8 inches and was made with a mesh drumhead to reduce the acoustic sound. The electronic

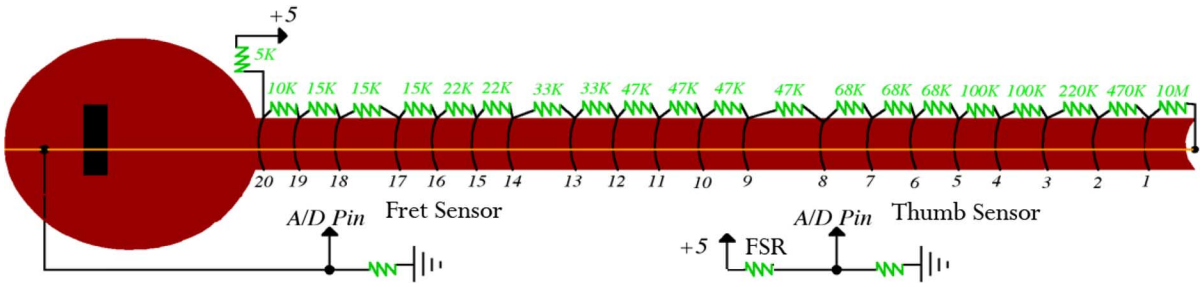

Fig. 4. E-Sitar circuit.

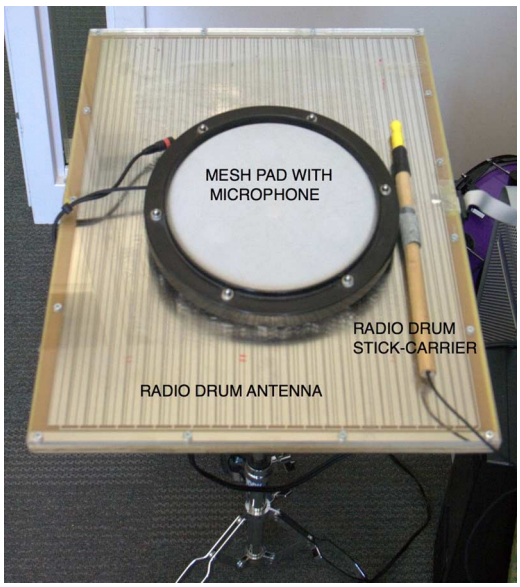

Fig. 5. Drum pad on radio drum surface.

drum pad is manufactured with a piezoelectric microphone attached to the underside of the head. The microphones on the drums were connected to a Mark of the Unicorn audio interface operating at CD quality sound (16-bits resolution with a 44 100 kHz sampling rate). The audio interface was connected to a computer running the analysis software.

The direct sensor used for training is the Radio Drum [26], which is based on capacitance sensors. It can detect the x,y,z positions of two drum sticks in 3-D space. This allowed us to place the surface of the Radio Drum under the snare drum or electronic drum pad and still be able to measure the stick position (see Fig. 5). Using the radio drum, quantized position was measured along the X and Y axes of the surface using 7-bits resolution and transmitted using the MIDI standard as integers between 0 and 127. For each test, the radio drum was calibrated to ensure proper accuracy. It is important to note that even though the training setup might require calibration, the trained "surrogate" sensor does not.

The electronic drum pad has a diameter of 8 inches (20.3 cm). The drum pad was placed in the center of the Radio Drum pad which returns approximately 20 values across that radius (10.17 cm) providing a measuring resolution of nearly 0.5 cm. The goal of the surrogate sensor is to provide the same resolution for estimating the drum strike position but only based on the analyzed acoustic output captured by the microphone.

C. Audio Feature Extraction

The feature set used in this paper is based on standard features used in isolated tone musical instrument classification, music, and audio recognition [27]. Our goal is not to find the optimal set of audio features for the proposed tasks. One of the nice properties of approaches for musical gesture acquisition that utilize machine learning compared to pure digital signal processing approaches is that the features utilized can be noisy, incomplete, redundant, and still provide useful information. Therefore, the features we use are standard and only slightly adapted for the particular problems we examine. We believe that our "surrogate" sensor approach can be used with any "reasonable" set of audio features.

Ideally the size of the analysis and texture windows should correspond as closely as possible to the natural time resolution of the gesture we want to map. In our experiments, we have looked at how these parameters affect the desired output. In addition, the range of values we explored was determined empirically by inspecting the data acquired by the sensors. The total latency of the system is determined by several factors, mainly the latency of audio input/output of the underlying operating system as well as the latency of the analysis window for feature extraction, and is typically in the range of 20 to 50 ms. Although this is adequate for many musical gestures of interest, there are cases where it would not be sufficient like the detection of fast drum hits. At the same time, this is an inherent limitation of any non-invasive audio-based approach.

For the E-Sitar experiments, it consists of four features computed based on the short time Fourier transform (STFT) magnitude of the incoming audio signal. It consists of the Spectral Centroid, Rolloff, and Flux as well as RMS energy which are described in more detail below. The features are calculated using a short time analysis window with duration 10–40 ms. In addition, the means and variances of the features over a larger texture window (0.2–1.0 s) are computed resulting in a feature set with 8 dimensions. The larger texture window captures the dynamic nature of spectral information over time, and it was a necessary addition to achieve better results in mapping features to gestures.

For the drum experiments the analysis window is 40 ms (no texture window) and the features used were: Spectral Centroid, Rolloff, Kurtosis, and Skewness as well as mel-frequency cepstrum coefficients (MFCCs). A preprocessing step of silence removal and onset detection ensure that features are only calculated once for each drum hit. The analysis window is located

so that it captures most of the energy of the hit. The Marsyas[1] audio analysis and synthesis framework was used for the feature extraction and direct sensor acquisition and alignment with the audio features [28].

The features calculated for each analysis window indexed by t are as follows.

1) Temporal Centroid: Temporal centroid is the center of gravity of the time domain representation of the signal as given by

$$Tc_t = \frac{\sum_{i=1}^{N} |x_i|}{n} \tag{1}$$

where x is the signal to be evaluated, N is the number of samples, and i is the number of samples to be evaluated.

2) RMS: RMS, root mean squared, is a measurement of amplitude that returns the value as given by

$$RMS_t = \sqrt{\frac{\sum_{i=1}^{N} x_i^2}{N}}. \tag{2}$$

See 1) for explanation of symbols.

3) Spectral Centroid: Spectral centroid returns the center of gravity of the magnitude spectrum as given by

$$SC_t = \frac{\sum_{k=0}^{N/2} k * |X_t(k)|}{\sum_{k=0}^{N/2} |X_t(k)|} \tag{3}$$

where $X(k)$ is the spectrum of the signal given by an FFT calculation, and N is the number of analysis frames (determined by FFT size).

4) Spectral Flux: Spectral flux measures the amount of local change over time in the frequency domain. It is defined by squaring the difference between normalized magnitudes in the frequency domain of frame t and $t-1$. If $N_t[n]$ and $N_t[n-1]$ are defined by the normalized spectrum magnitude of frame t and $t-1$, then the spectral flux F_t is given by

$$F_t = \sum_{k=1}^{N} \left(N_t[k] - N_{t-1}[k]\right)^2. \tag{4}$$

It should be noted that magnitudes are normalized by dividing each value in every frame by the RMS value of that frame [29]. F_t is calculated for each frame and then averaged over time in order to yield one value for spectral flux.

5) Spectral Rolloff: Spectral rolloff is another feature that describes the spectral shape [29]. It is defined as the frequency R_t below which 85% of the magnitude of the spectrum is concentrated. If $M_t[n]$ is the magnitude of the spectrum, then the spectral rolloff $R_t$is given by

$$\sum_{k=1}^{R_t} \left(M_t[k]\right) = .85 * \sum_{n=1}^{N} \left(M_t[k]\right). \tag{5}$$

[1]http://marsyas.sourceforge.net.

6) Spectral Skewness: Spectral skewness is a third-order moment that returns the skewness of the spectrum as given by

$$Sk_t = \frac{\sum_{k=1}^{N} \left(M_t[k] - u\right)^3}{\sigma^3} \tag{6}$$

where x is the magnitude of the spectrum of the signal, u is the mean of the signal, and σ is the spectrum distribution standard deviation.

7) Spectral Kurtosis: Spectral kurtosis is a fourth-order moment that examines how outlier prone the spectrum is. A spectrum with normal distribution will have a spectral kurtosis of 3. The function in this experiment conforms to the convention where three is subtracted from the kurtosis so that a spectrum with normal distribution will have a spectral kurtosis of 0:

$$K_t = \left(\frac{\sum_{k=1}^{N} \left(M_t[k] - u\right)^4}{\sigma^4}\right) - 3. \tag{7}$$

8) Mel-Frequency Cepstrum Coefficients: MFCC are a product of two distinct stages of operations. First, the cepstrum of the signal is calculated, which is given by taking the log of the magnitude spectrum. This effectively smooths the spectral content of the signal. Second, the spectrum is divided into 13 bands based on the mel scale, which is a scale based on human perception of pitch [30].

This feature returned a set of coefficients for each FFT frame of the signal that was analyzed. A 256-point FFT size was used providing 13 coefficients for each FFT frame.

D. Classification and Regression

Classification refers to the prediction of discrete categorical outputs from real-valued inputs. A variety of classifiers have been proposed in the machine learning literature [8] with different characteristics in respect to training speed, generalization, accuracy, and complexity. The main goal of the paper is to provide evidence to support the idea of using direct sensors to train surrogate sensors in the context of musical gesture detection. Therefore, experimental results are provided using a few representative classification methods. Regression refers to the prediction of real-valued outputs from real-valued inputs. Multivariate regression refers to predicting a single real-valued output from multiple real-valued inputs. A classic example is predicting the height of a person using their measured weight and age. There are a variety of methods proposed in the machine learning [8] literature for regression. Ordinal regression is a specialized form of regression where the predicted output consists of discrete labels that are ordered. For example, when predicting the strike position in relation to the center of a drum, it can be either a continuous value (regression) or an ordinal value with values such as center, middle, and edge (ordinal regression). Ordinal regression problems can be treated as classification problems that do not assume order among the labels, but there are also specialized techniques.

For some of the experiments described below, we use linear regression where the output is formed as a linear combination

of the inputs with an additional constant factor. Linear regression is fast to compute and therefore useful for doing repetitive experiments for exploring different parameter settings. We also employ a more powerful back propagation neural network [8] that can deal with nonlinear combinations of the input data. The neural network is slower to train but provides better regression performance. Finally, the M5 prime decision tree-based regression algorithm was also used [31]. The performance of regression is measured by a correlation coefficient which ranges from 0.0 to 1.0 where 1.0 indicates a perfect fit. In the case of gestural control, there is significant amount of noise and the direct sensor data does not necessarily reflect directly the gesture to be captured. Therefore, the correlation coefficient can mainly be used as a relative performance measure between different algorithms rather than an absolute indication of audio-based gestural capturing. The automatically annotated features and direct sensor labels are exported into the Weka[2] machine learning framework for training and evaluation [32]. For evaluation and to avoid over-fitting the surrogate sensors, we employ a 50% percentage split where half the collected data is used for training and the remaining is used for testing. This ensure pairs of correlated feature vectors that are close together in time do not get split into training and testing.

V. Experimental Results

In this section, we present experimental results of how the idea of surrogate sensors can be used in the context of musical gesture acquisition in two concrete case studies: predicting from the audio signal thumb pressure and fret location in the E-Sitar as well as the strike position in an acoustic snare drum and electronic drum pad. Although a "surrogate" sensor setup requires much less manual involvement than audio annotation, it still takes some musician time to train. In the following experiments, we have chosen to utilize a reasonable amount of training data that provides good performance without tiring out the performer.

A. E-Sitar Results

The goal of the experiments with the E-Sitar was to explore the idea of using "surrogate" sensors for capturing the fret and thumb data for sitar performance. We show results from two experiments. The first experiment used limited data, a single player, and a subset of the audio features described above and is reproduced from [23]. Although our current version achieves slightly better results than the ones reported in our previous work [23] for the first experiment, we still report the previous results as the conclusions about the choice of parameters remain the same.

For the second experiment, three sitar players performed two sets of data. Our first data set was designed to record a player's individual performance characteristics during disciplined practice exercises. We chose two central exercises from the vast literature of classical North Indian practice methods [33]: Bol patterns and Alankars. Bol patterns are specific patterns of da (up stroke), ra (down stroke), and diri (up stroke and then down stroke in rapid succession), which are explicitly used in sitar practice/training, as well as in performance. Alankars refer to scalar patterns that can be modally transposed; they form the basis of many musical ornaments and are also often used for melodic development. For our second data set, each performer played a fixed composition. As in the exercises, the composition makes specific use of both the left and right hands, but with more room for ornamentation, microtiming, and other expressive nuances. For the experiments reported in this paper, both datasets were combined. All data from all sensors are sampled at 100 Hz and stored as uncompressed wav files at a 44 100 Hz sampling rate. A metronome was also used, allowing for more highly controlled and synchronous experiment set up.

[2]http://www.cs.waikato.ac.nz/ml/weka/.

TABLE I
Effect of Analysis Window Size

Analysis Window Size	128	256	512
Correlation Coefficient	0.2795	0.3226	0.2414

TABLE II
Regression on Sitar Thumb Data

Regression Type	Player 1	Player 2	Player 3	All
Linear Regression	0.6427	0.4054	0.4155	0.585
SMO	0.6051	0.3834	0.3714	0.5125
Pace	0.6417	0.399	0.4122	0.3824
Neural Network	0.8973	0.6387	0.9415	0.9231
M5' Regression	0.8706	0.6387	0.9151	0.8989

Our first experiment was to analyze the effect of the analysis window size used for audio feature extraction for predicting thumb pressure from audio analysis of the microphone input. Table I shows the results. The texture size remained constant at 0.5 s and linear regression was used. The correlation coefficient for random inputs is 0.14. It is apparent based on the table that an analysis window of length 256 (which corresponds to 10 ms) achieves the best results. It can also be seen that the results are significantly better than chance. We used this window size for all the following experiments. The low correlation scores are due to smaller amounts of training data and a reduced feature set used in the initial conference paper [23].

Table II shows the correlation coefficients for different types of regression algorithms for predicting thumb pressure from acoustic analysis of the microphone input. These results have not been reported previously. The obtained correlation coefficients are quite good especially for certain combinations of algorithms and players. The last row shows the results of using data from all three players and indicates that the trained "surrogate" sensors can be generalized to more than one player without significantly losing classification accuracy. It is important to note that in most cases, we are interested in derivative information from the "surrogate" sensor such as detecting up-strokes and down-strokes. Therefore, even lower correlation coefficients are adequate for our purposes.

Table III shows the correlation coefficients for different types of regression algorithms for predicting the fret from acoustic analysis of the microphone input. These results have not been reported previously. The obtained correlation coefficients are quite good especially for certain combinations of algorithms and players. This is a particularly interesting example as it essentially performs a form of discrete pitch detection based on a

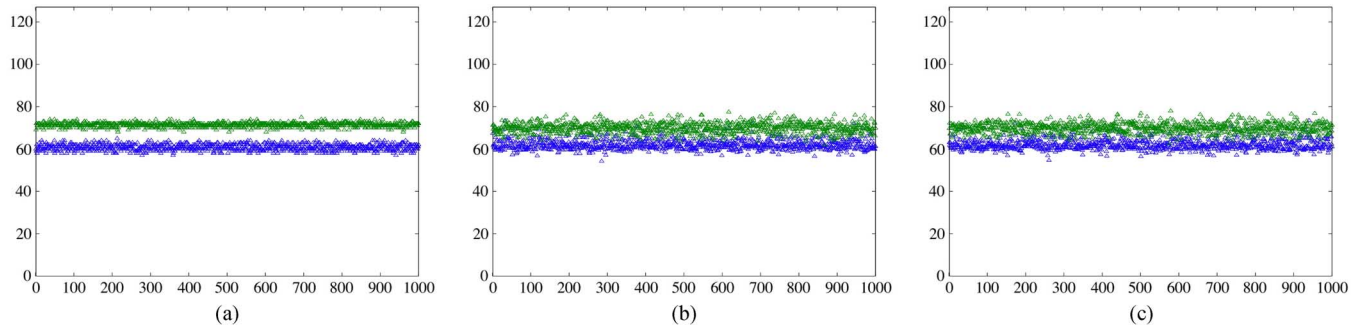

Fig. 6. Regression results for predicting drum strike position using a surrogate sensor. The x-axis is the strike index and the y-axis is the predicted regression output corresponding to distance from the center scaled to return values in the same range as the radio drum. (a) RadioDrum input. (b) Surrogate sensor. (c) Surrogate sensor with discrete classes.

TABLE III
REGRESSION ON SITAR FRET DATA

Regression Type	Player 1	Player 2	Player 3
Linear Regression	0.7058	0.652	0.5888
SMO	0.6851	0.6245	0.5578
Pace	0.706	0.6542	0.5858
Neural Network	0.9367	0.8541	0.9094
M5' Regression	0.8624	0.8739	0.8734

TABLE IV
REGRESSION USING OTHER PLAYERS FOR TRAINING SET

Regression Type	Player 1	Player 2	Player 3
Linear Regression	0.6448	0.4139	0.4342
SMO	0.5839	0.3745	0.3923
Pace	0.6446	0.4134	0.4343
Neural Network	0.8906	0.7415	0.9546
M5' Regression	0.9572	0.8897	0.9856

supervised learning without any "prior" knowledge about what pitch is.

Table IV shows the correlation coefficients where each classifier is trained on the data of two players and used to predict the sensor data of the remaining player. This form of three-fold cross-validation demonstrates that "surrogate" sensors generalize across different players and is not tied to a specific performer. Each classifier receives over 10 000 feature vectors to train. The best results of the study are shown in this graph with the M5 classifier on player 3 achieving a correlation coefficient of 0.9856.

B. E-Snare

The third author completed a Master's thesis [34] on the topic of indirect acquisition of snare drum gestures. In this thesis, 1260 samples were collected with three drums and three expert players. The process of collecting and processing the training data took nearly a week of manual labor. Using the method described in this paper, the same process took under an hour.

A classically trained percussionist was used for data collection, and no pre-processing or post-processing of the classification results was performed. In each of the experiments, unless explicitly mentioned, the hits were regularly spaced in time. For each hit, the radial position was measured and the hit was labeled as either "edge" or "center" using thresholding of the Radio Drum input. Audio features are also extracted in real-time using input from a microphone. The features and sensor measurements are then used for training classifiers. The setup can be viewed in Fig. 5.

TABLE V
PERCENTAGES OF CORRECTLY CLASSIFIED DRUM PAD HITS (CENTER, HALFWAY, OR EDGE)

	Classifier	**Ordinal**
ZeroR	36.5285	NA
SMO	80.3109	76.1658
Naive Bayes	76.6839	76.1658
J48	86.2694	89.1192
KNN	88.342	88.342
Neural Network	88.8601	90.4145

In the first experiment, the electronic drum pad was hit in the center and at the edge. One thousand samples of each strike location were captured and used for classification. Fig. 6(a) shows a graph of the MIDI data captured by the Radio Drum for each strike. Fig. 6(b) shows a graph of the predicted output from a PACE regression classifier. The result was a correlation coefficient of 0.8369 with an absolute error of 2.3401 and a mean squared error of 2.9784. The graph clearly shows enough separation between the two classes. The data was then divided into two symbolic classes: Center and Edge. The data was run through the PACE regression classifier using the mean of the Radio Drum input for each class. The results were slightly improved—a correlation coefficient of 0.8628 with an absolute error of 2.0303 and a mean squared error of 2.6758.

The error achieved in the regression tests suggests that the algorithm has an accuracy of approximately 1 cm. Each MIDI value provided by the Radio Drum corresponds to approximately 0.5 cm and with an error of approximately 2, depending on the algorithm, this leads to a worst-case error of 1 cm. Therefore, even though the trained "surrogate" is not as accurate as the Radio Drum input, it still provides enough resolution to discriminate between center and edge easily.

Table V shows classification results for predicting whether a mesh electronic drum pad was hit in the center, halfway, or the edge. As can be seen, excellent classification results can be obtained using the surrogate sensor approach. A total of 348 drum hits were used for this experiment.

Table VI shows classification results for predicting whether an acoustic snare drum was hit in the center or the edge. The **Snares, No Snares** rows are calculated using approximately

TABLE VI
PERCENTAGES OF CORRECTLY CLASSIFIED SNARE DRUM HITS

	ZeroR	NB	MLP	MLR	SMO
Snares	53	92	91	91	92
No Snares	57	93	94	95	95
Improvisation	59	79	77	78	78

TABLE VII
RADIO DRUM REGRESSION FROM WITH 1057 INSTANCES MOVING FROM CENTER TO EDGE

Regression Type	Correlation Coefficient
Linear Regression	0.4594
SMO	0.6522
Pace	0.6579
Neural Network	0.6737
M5' Regression	0.7394

1000 drum hits with the snares engaged/not engaged. All the results are based on ten-fold cross-validation. The trivial *ZeroR* classifier is used as a baseline. The following classifiers are used: *Naive Bayes* (NB), *Multi-Layer Perceptron* (MPL), *Multinomial Logistic Regression* (MLR), and *Support Vector Machine* trained using sequential minimal optimization (SMO). The results are consistent between different classifier types and show that indirect acquisition using audio-based features trained using direct sensors is feasible. The **Improvisation** row is calculated using 200 drum hits of an improvisation rather than the more controlled input used in the other cases where the percussionist was asked to alternate regularly between hitting the edge and the center of the drum. Even though the results are not as good as the cleaner previous rows, they demonstrate that any performance can potentially be used as training data. The main reason that the results are lower in the improvisation case is that there is more noise in the ground truth acquired by the radio drum sensors as the player is less precise when hitting the drum. The use of patterned input constraints the performer to some extent as it requires a specific calibration phase but has the potential of improved performance. In practice, we have used both approaches depending on the specific requirements of the particular music performance.

Ordinal regression [35] was computed for all tests to evaluate any difference. Tracking of strike position is a candidate for ordinal regression because the classes are ordered. Marginal improvements on some classifiers were obtained when ordinal regression was applied (see Table V).

An experiment was conducted to train a regression classifier using the Radio Drum as the direct sensor. Data was collected by playing on the drum moving gradually from edge to center and back to edge for a total of 1057 strikes (see Table VII). This experiment illustrates the "surrogate" sensor in the intended application of rapid data collection and training of a classifier.

To verify the effectiveness of the features used for classification, an experiment was conducted to progressively add features. The feature vector was reduced to one element and then increased until all 17 features were included (see Fig. 7). The plot shows an increasing line as features are added back into the vector and the correlation coefficient increases.

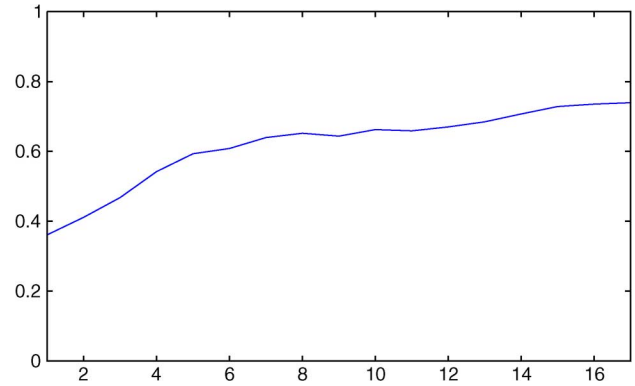

Fig. 7. Effect of more features to the correlation coefficient in drum regression. The y-axis is the correlation coefficient and the x-axis is the discrete feature index.

VI. DISCUSSION AND FUTURE WORK

In this paper, we apply the concept of a surrogate sensor to "train" machine learning model based on audio feature extraction for indirect acquisition of music gestures. Once the model is trained and its performance is satisfactory, the direct sensors can be discarded. Large amounts of training data for machine learning may be collected with minimum effort just by playing the instrument. In addition, the learned indirect acquisition method allows capturing of nontrivial gestures without modifications to the instrument. We believe that the idea of using direct sensors to train indirect acquisition methods can be applied to other area of interactive media and data fusion.

In the future, more features will be added to the system and a study of the effectiveness of various features will be conducted. We also plan to explore the application of the "surrogate" sensor concept to other musical instrument gesture acquisition scenarios. Two specific examples we plan to explore are detection of string played in the violin and of type of mouthpiece in woodwinds. In both cases, both direct sensing approaches as well as indirect audio-based approaches have been proposed in the literature and can be combined using a surrogate sensor approach.

Creating tools for further processing the gesture data to reduce the noise and outliers is another direction for future research. Another eventual goal is to use these techniques for transcription of music performances. Currently, this system is used regularly in performance by the first two authors.

ACKNOWLEDGMENT

The authors would like to thank M. Wright and J. Hochenbaum for providing additional data for the E-Sitar experiments.

REFERENCES

[1] T. Machover, Hyperinstruments: A Progress Report, MIT, 1992, Tech. Rep.
[2] A. Kapur, P. Davidson, P. Cook, P. Driessen, and A. Schloss, "Digitizing North Indian performance," in *Proc. Int. Computer Music Conf. (ICMC)*, Miami, FL, 2004.
[3] D. Young and I. Fujinaga, "Aobachi: A new interface for Japanese drumming," in *Proc. New Interfaces for Musical Expression (NIME)*, Hamamatsu, Japan, 2004.

[4] M. M. Wanderley and P. Depalle, "Gestural control of sound synthesis," *Proc. IEEE*, vol. 92, no. 4, pp. 632–644, Apr. 2004.
[5] O. Vallis, J. Hockenbaum, and A. Kapur, "Extended interface solutions for musical robotics," in *Proc. IEEE Int. Symp. Multimedia*, 2008.
[6] C. J. M. Gimenes and E. R. Miranda, "Musicianship for robots with style," in *Proc. New Interfaces for Musical Expression*, 2007.
[7] E. R. Miranda and M. Davy, Eds., New Digital Musical Instruments: Control and Interaction Beyond the Keyboard A-R Edition, 0-89579-585-X, 2006.
[8] T. Mitchell, *Machine Learning*. Columbus, OH: McGraw-Hill, 1997.
[9] E. B. Egozy, "Deriving musical control features from a real-time timbre analysis of the clarinet," Master's thesis, Massachusetts Institute of Technology, Cambridge, 1995.
[10] N. Orio, "The timbre space of the classical guitar and its relationship with plucking techniques," in *Proc. Int. Computer Music Conf. (ICMC)*, 1999.
[11] C. Traube and J. O. Smith, "Estimating the plucking point on a guitar string," in *Proc. Conf. Digital Audio Effects*, 2000.
[12] C. Traube, P. Depalle, and M. Wanderley, "Indirect acquisition of instrumental gestures based on signal, physical and perceptual information," in *Proc. Conf. New Musical Interfaces for Musical Expression*, 2003, pp. 42–47.
[13] F. Gouyon and P. Herrera, "Exploration of techniques for automatic labeling of audio drum tracks' instruments," in *Proc. MOSART: Workshop Current Directions in Computer Music*, 2001.
[14] J. Silpanpää, Drum Stroke Recognition Tampere University of Technology, Tampere, Finland, 2000, Tech. Rep. [Online]. Available: www.cs.tut.fi/sgn/arg/music/drums/raportti.ps.
[15] A. Tindale, A. Kapur, G. Tzanetakis, and I. Fujinaga, "Retrieval of percussion gestures using timbre classification techniques," in *Proc. Int. Symp. Music Information Retrieval*, 2004.
[16] A. Kapur, G. Tzanetakis, and P. F. Driessen, "Audio-based gesture extraction on the esitar controller," in *Proc. Conf. Digital Audio Effects*, 2004.
[17] S. Kabadayi, A. Pridgen, and C. Julien, "Virtual sensors: Abstracting data from physical sensors," in *Proc. Int. Workshop Wireless Mobile Multimedia*, 2006, pp. 587–592.
[18] D. King, W. Lyons, C. Flanagan, and E. Lewis, "An optical-fiber sensor for use in water systems utilizing digital signal processing techniques and artificial neural network pattern recognition," *IEEE Sensors J.*, vol. 4, no. 1, pp. 21–27, 2004.
[19] D. Prokhorov, "Virtual sensors and their automotive applications," in *Proc. Sensor Networks and Information Processing Conf.*, 2005, pp. 411–416.
[20] B. Krishnapuram, J. Sichina, and L. Carin, "Physics-based detection of targets in SAR imagery using support vector machines," *IEEE Sensors J.*, vol. 3, no. 2, pp. 147–157, 2003.
[21] E. Hanzevack, T. Long, C. Atkinson, and M. Traver, "Virtual sensors for spark ignition engines using neural networks," in *Proc. Amer. Control Conf.*, 1997, vol. 1, pp. 669–673.
[22] K. Marko, J. James, T. Feldkamp, G. Puskorius, and L. Feldkamp, "Signal processing by neural networks to create virtual sensors and model-based diagnostics," in *Proc. Artificial Neural Networks: ICANN 96: 1996 Int. Conf.*, Bochum, Germany, Jul. 16–19, 1996, p. 191, Springer.
[23] A. Kapur, G. Tzanetakis, and P. F. Driessen, "Audio-based gesture extraction on the esitar controller," in *Proc. Conf. Digital Audio Effects*, 2004, pp. 17–21.
[24] A. Kapur, G. Tzanetakis, and A. R. Tindale, "Learning indirect acquisition of instrumental gestures using direct sensors," in *Proc. IEEE Workshop Multimedia Signal Processing*, 2006, pp. 37–40.
[25] S. Bagchee, *Understanding Raga Music*. Mumbai, India: Ceshwar Business, 1998.
[26] M. Mathews and W. Schloss, "The radio drum as a synthesizer controller," in *Proc. Int. Computer Music Conf. (ICMC)*, Columbus, OH, 1989.
[27] , A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. New York: Springer-Verlag, 2006.
[28] G. Tzanetakis, "Marsyas: A case study in implementing music information retrieval systems," in *Intelligent Music Information Systems: Tools and Methodologies*. Hershey, PA: Information Science Reference, 2008, pp. 31–49.
[29] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, 2002.
[30] B. Logan, "Mel-frequency cepstrum coefficients for music modeling," in *Proc. Int. Symp. Music Information Retrieval*, 2000.
[31] G. Holmes, M. Hall, and E. Frank, "Generating rule sets from model trees," in *Proc. 12th Australian Joint Conf. Artificial Intelligence*, 1999, pp. 1–12, Springer-Verlag.
[32] I. Witten, E. Frank, and M. Kaufmann, *Data Mining: Practical Machine Learning Tools With Java Implementations*. San Francisco, CA: Addison-Wesley, 2000.
[33] A. Kahn and G. Ruckert, *The Classical Music of North India*. New Delhi, India: Munshiram Manoharlal, 1998.
[34] A. Tindale, "Classification of snare drum sounds using neural networks," Master's thesis, McGill University, Montreal, QC, Canada, 2004.
[35] E. Frank and M. Hall, "A simple approach to ordinal classification," in *Proc. 12th Eur. Conf. Machine Learning*, Sep. 5–7, 2001, pp. 145–156.

Adam Tindale received the B.Mus. degree from Queen's University, Kingston, ON, Canada, in 2001, the M.A. degree in music technology from McGill University, Montreal, QC, Canada, in 2004, and the Interdisciplinary Ph.D. degree in music, computer science, and electrical engineering at the University of Victoria, Victoria, BC, Canada.

He is currently a Permanent Instructor of Interaction Design in the Media Arts and Digital Technologies area at the Alberta College of Art and Design, Calgary, AB, Canada. His research interests include indirect acquisition of percussive gestures, music technology in education, assistive technology, and musical applications of machine learning techniques.

Ajay Kapur received the B.S.E. degree in computer science from Princeton University, Princeton, NJ, in 2002 and the Interdisciplinary Ph.D. degree from the University of Victoria, Victoria, BC, Canada, in 2007.

He is the Director of Music Technology at California Institute of the Arts and founder of KarmetiK, a set of musicians, scientists, and artists who combine Indian Classical music with modern technology.

George Tzanetakis (M'03) received the Ph.D. degree in computer science from Princeton University, Princeton, NJ, in 2002.

He is an Associate Professor in the Department of Computer Science with cross-listed appointments in Electrical and Computer Engineering and Music at the University of Victoria, Victoria, BC, Canada. He was a Post-Doctoral Fellow at Carnegie Mellon University, Pittsburgh, PA, in 2002–2003. His research spans all stages of audio content analysis such as feature extraction, segmentation, and classification, with specific emphasis on music information retrieval. He is also the primary designer and developer of Marsyas, an open source framework for audio processing with specific emphasis on music information retrieval applications. More recently, he has been exploring new interfaces for musical expression, music robotics, computational ethnomusicology, and computer-assisted music instrument tutoring. These interdisciplinary activities combine ideas from signal processing, perception, machine learning, sensors, actuators, and human-computer interaction with the connecting theme of making computers better understand music to create more effective interactions with musicians and listeners.

Dr. Tzanetakis received a IEEE Signal Processing Society Young Author Award for his pioneering work on musical genre classification, which is frequently cited.