Real-time Recognition of Guitar Performance Using Two Sensor Groups for Interactive Lesson

Yejin Shin, Jemin Hwang, Jeonghyeok Park, Soonuk Seol
School of Electrical, Electronics & Communication, Korea University of Technology & Education
Cheonan, South Korea
{yepp1252, zpdltusl, 2011161058, suseol}@koreatech.ac.kr

ABSTRACT
The accurate recognition of guitarist performance is challenging compared with other instruments because a guitar player typically plays several notes at once and uses both hands in different ways. In this paper, we propose a sensor-based guitar that consists of two groups of sensors. One sensor is used to recognize the fingering positions of the fretting hand, and the other is used to detect the guitar strings that are played by the picking hand. We design an embedded system for accurate sensing and propose a data analysis mechanism to precisely figure out the played pitch and the duration of notes using the sensed data. We realize our scheme as a high-quality prototype that detects guitarist performance with accuracy sufficient for the transcribing of a performance. We also present real application examples such as a rhythm game for interactive lessons and a music sharing feature with user created musical scores.

Author Keywords
Music transcription; automatic note inference; guitar; sensors; fingering position detection; performance feedback

ACM Classification Keywords
• *Applied computing~Sound and music computing*
• *Applied computing~Interactive learning environments*
• *Hardware~Tactile and hand-based interfaces*

INTRODUCTION
With the development of ICT and multimedia technology, many fields are changing. Things that were unimaginable in the past, even things unrelated to IT, are now connected to the Internet, and various data analyses are becoming a driving force of the changes in society.

Convergence of musical instruments and IT is one such trend. The development of an IT guitar to assist beginning guitar players is one example of this trend. These products mainly focus on providing a guide, for fingering or playing technique, on the fretboard by using embedded LEDs. Fretlight [4] is a product that uses LEDs installed on the fretboard of the guitar to provide a fingering position guide by turning on the LEDs. Fret Zealot [5] is another product, comprising an LED strip and controller attached to the guitar. Still, existing products do not provide feedback through identification of correct playing. Therefore, research into providing feedback on user's playing through analysis of multimedia data from musical instruments has conducted.

A number of techniques are being developed to sense the performance of musical instruments to transcribe musical scores or give feedback to players. Techniques for analyzing audio by recording sounds from instruments [1,3,17,18] have been traditionally widely studied. However, there is a limitation in that the audio-based approach is applicable to fixed-tuned musical instruments. In addition, it is difficult to estimate pitches when multiple sounds are played at the same time, such as a chord, and even harder when there is noise from the microphone used. Another approach proposed in [19] analyzes violin fingering through video analysis. However, the match ratio for the onset, offset, and pitches elements is only 65%, which is limited for practical applications. Moreover, a string instrument has the characteristic that even if the fingerboard is fingered, it does not make a sound unless the strings are actually plucked. Therefore, judging the onset and offset depending on whether the fingerboard is fingered may result in a difference from the actual performance.

To increase sensing accuracy, approaches that involve physically embedded sensing circuits in musical instruments have been studied. Pianos with embedded sensors [6,10] are such studies. However, these pianos require only one kind of sensor and thus detecting a key press is relatively simple. More complicated approaches are required in the case of string instruments. One study [14] proposed an approach to judge the accuracy of a musical performance on a cello. The approach includes a sensing circuit on the fingerboard and combines sound recording data through a microphone and motion sensing data via a video. Other approaches [2,7,8,12] involved attaching sensors to the fingerboard and bow of a violin to sense the fingering and bowing of the strings. However, such approaches are difficult to apply to guitars. A guitar is played by fingering the chord with one hand and plucking

TEI '18, March 18–21, 2018, Stockholm, Sweden

ACM ISBN 978-1-4503-5568-1/18/03…$15.00
https://doi.org/10.1145/3173225.3173235

the strings with the other hand or a pick. Therefore, both the left and right hand motions, which cause the vibrations of the strings, should be sensed to recognize guitar performance. Similar to a sensor-attached bow in the case of a violin, a sensor circuit can be configured by wiring a metal guitar pick [15,16] However, this approach causes discomfort to players and limits guitarists from playing in various styles (e.g., fingerstyle), particularly in acoustic guitar. These difficulties have imposed limitations on transcribing music from an acoustic guitar and eliminating the gaps that have to be overcome to actually apply it to real instruments.

In this paper, we propose a sensor-based guitar. We add sensors to an LED-based guitar [13] that allows guitar students to practice guitar by lighting LEDs on the corresponding positions on the guitar fingerboard. Our sensor-based guitar consists of two groups of sensors. One sensor is used to recognize the fingering positions of the fretting hand, and the other is used to detect the guitar strings that are played by the picking or plucking hand. We design an embedded system for accurate sensing and propose a data analysis mechanism to precisely figure out the played pitch and the duration of notes by using the sensed data. The proposed method realizes music transcription with a very high level of accuracy through a sensor- and circuit-based approach. We realize our idea as a high-quality prototype and conduct experiments to evaluate the accuracy of the detection. In the evaluation, we analyze and compare the sound recorded in audio and the data sensed in the prototype. To the best of our knowledge, the proposed guitar is the first attempt to accurately recognize performance information by sensing both the fingered position and plucked strings when playing a guitar. We also present real application examples such as a rhythm game for interactive lessons and a music sharing feature with user created musical scores.

DESIGN AND IMPLEMENTATION OF SENSING CIRCUIT

Figure 1. Sensor-based guitar

We design sensing circuits and implement a commercial-grade guitar prototype by integrating the sensing circuits. The guitar is composed of a main controller as an embedded system, a fingerboard-sensing circuit, and a string-sensing circuit. The external appearance of the guitar is shown in Fig. 1. We implement an embedded system using Atmel SAM3A8C as a Micro Controller Unit (MCU), which is connected to the fingerboard-sensing circuit and string-sensing circuit.

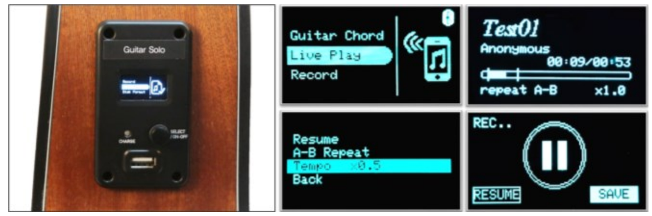
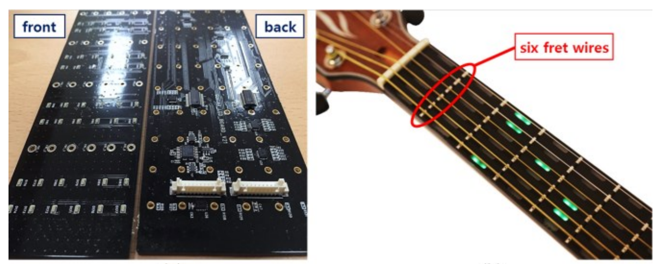


Figure 2. Various features of the embedded system

The embedded system has been significantly improved to adapt sensing features while providing all the existing useful features from our previous study [13]. For example, with the LED guidance feature, a user can easily find fingering positions on the fingerboard. The user can also search for songs from a music server and download them to the embedded system through a smartphone application. Then, without a smartphone, the user can practice the stored songs with LED guidance. In addition, the user can quickly master a certain song by using the efficient practice features, such as Rewind, Fast-Forward, and AB Repeat, provided in the embedded system. By newly employing sensors, the system can recognize the pitches and duration of the notes played. Fig. 2 shows the embedded system and some displayed menus. In this paper, we focus on the design and implementation of the fingerboard-sensing circuit and string-sensing circuit.

Fingerboard-Sensing Circuit



(a) (b)

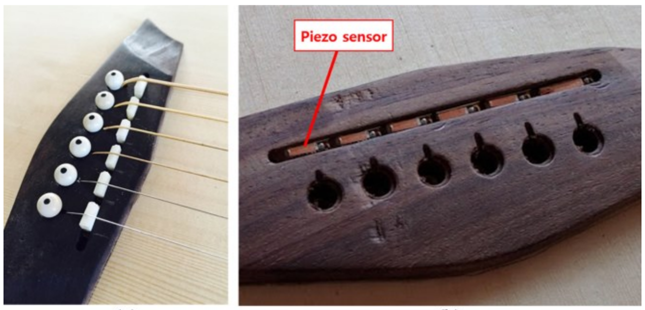
Figure 3. The fingerboard of the sensor-based guitar: (a) PCB embedded under the fingerboard, (b) the fingerboard with fret wires

To detect fingering positions, we have integrated a sensing circuit with an LED circuit, as shown in Fig. 3(a), and placed it under the fingerboard, as shown in Fig. 3(b). We have split a conventional fret wire into six fret wires, which are parts of the sensing circuit. This allows a total of 120 fret wires to play a role as individual switches when they are in contact with conductive strings.

We use each string as a part of the circuit so that if a particular string and fret wire touch, a closed loop between the fingerboard and the string is made and the position can be electrically recognized. Information on simultaneous multiple positions is input into the main controller in the form of a bit stream. By using 16 8-bit shift registers, it is possible to sense various combinations of fingering information for 120 fret wires with only six data input pins.

String-Sensing Circuit

The guitar employs six piezo sensors to recognize whether the strings have been plucked. The piezo sensors are located under the saddles, which sense the change of vibration. However, to prevent the vibrations from transmitting to other piezo sensors through the saddle, we divide the saddle into six and attach the piezo sensors under each separated saddle, as shown in Fig. 4.



(a) (b)

Figure 4. The bridge of a sensor-based guitar: (a) six separate saddles, (b) six piezo sensors located under the saddles

The six sensors are connected to the analog input pins of the embedded system. The string sensing is determined by comparing an analog input value with a certain threshold value. Details on how to determine the duration of a note played are explained in the following section.

SENSING DATA ANALYSIS MECHANISM

When a user plays the guitar, recognizing the user's performance is accomplished through two sensing mechanisms. Fret sensing recognizes the fingering positions. String sensing recognizes the onset and offset for each string that is plucked. The sensed data are further processed by the embedded system to obtain meaningful information. In this section, we explain the mechanisms of analyzing the sensed data.

Fingering Position Sensing

Because we use each string as part of the circuit, if a particular string touches a particular fret wire, the MCU reads a high value for the fret wire; otherwise, it reads a low value. The MCU scans each fret wire every 2ms and finds fingering positions as a set of coordinates that express string and fret numbers. Therefore, even if multiple frets are fingered at once, the fingering position can be recognized with high accuracy. In a stringed instrument, if a user plucks a string while pressing two or more frets on the same string, the actual sound is generated by the vibration of a string length from the nearest fret to the guitar body to the saddle in the guitar body. Therefore, we consider the nearest fret to the saddle a meaningful fingering fret during the data-processing step shown in Fig. 5.

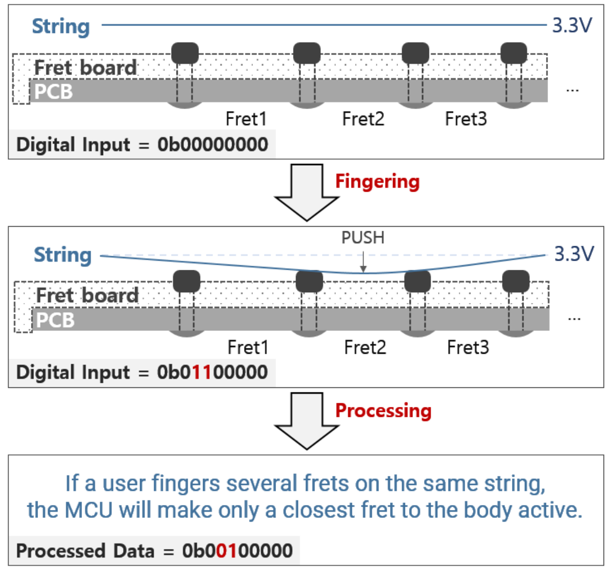


Figure 5. Fret-sensing mechanism

Recognizing the duration of a note

This section describes how to process the sensed data to recognize the duration of notes played by the user. We calculate the envelope of the analog data and compare it with the threshold to recognize the onset and offset of a note. Table 1 explains variables used for recognizing the duration of a note.

Envelope calculation mechanism

The MCU reads analog input data from the piezo sensors every 2ms ($Sample\ Rate = 500\ Hz$). The sample rate can be increased with a higher performance MCU. When a user plucks a string, the raw input data (V_{raw}) gradually decreases in amplitude, repeating positive and negative values. Therefore, if the raw data itself is used in the sensing mechanism, it is recognized that the string is plucked several times because V_{raw} repeatedly crosses the threshold. Therefore, we do not use V_{raw} directly, but calculate $V_{shifted}$ by $|V_{raw} - V_{baseline}|$, where the $V_{baseline}$ is the measured as an average vibration value of the string in the absence of any vibration. We calculate an envelope ($V_{envelope}$) of $V_{shifted}$ and compare it with the threshold to recognize the onset and offset of the note. The following equations are used to calculate the envelope.

$$g_{Attack} = e^{-\frac{1}{SampleRate \times t_{Attack}}} \tag{1}$$

$$g_{Release} = e^{-\frac{1}{SampleRate \times t_{Release}}} \tag{2}$$

$$V_{envelope(n)} = g \times V_{envelope(n-1)} + (1-g) \times V_{shifted(n)}$$
$$\begin{cases} g = g_{Attack}, V_{shifted(n)} > V_{envelope(n-1)} \\ g = g_{Release}, V_{shifted(n)} \le V_{envelope(n-1)} \end{cases} \tag{3}$$

Variables	Descriptions
V_{raw}	Raw vibration data from piezo sensor.
$V_{baseline}$	Average V_{raw} value when there is no vibration.
$V_{shifted}$	Vibration value shifted by baseline, i.e., $\lvert V_{raw} - V_{baseline}\rvert$.
$V_{envelope}$	Vibration value from the envelope of the oscillating signal.
t_{Attack}	Time for initial run-up of vibration level from nil to peak, beginning when the string is first plucked.
$t_{Release}$	Time for the level to decay from peak to zero.
g_{Attack}	Gain coefficient for attack time calculated by the time-constant equation, Eq. (1).
$g_{Release}$	Gain coefficient for release time calculated Eq. (2).
Th_{onset}	Threshold value of the envelope to determine onset.
Th_{offset}	Threshold value of the envelope to determine offset.
I_{max}	Maximum value of the influence on the envelope due to plucking other strings.

Table 1. The variables used for calculating the duration of a note

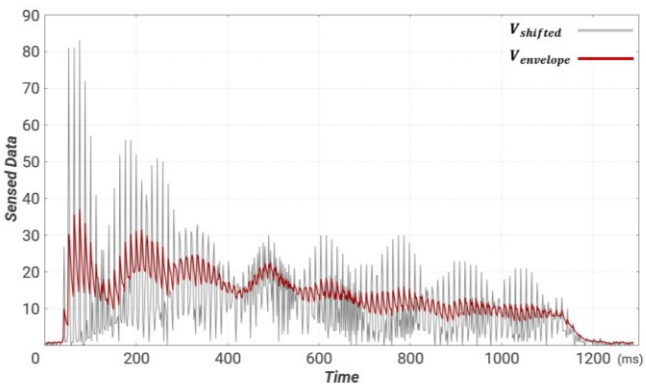


Figure 6. $V_{shifted}$ vibration waveform and $V_{envelope}$ calculation result

t_{Attack} in Eq. (1) is the time from when the sound begins to emerge until it reaches the peak, and it allows the user to specify how quickly the envelope follows a higher value. In contrast, $t_{Release}$ in Eq. (2) represents how quickly the envelope follows a smaller value. g_{Attack} and $g_{Release}$ are the gain factors that determine the rate of reflection and retention of the sampling data, respectively. Eq. (3) indicates that $V_{shifted(n)}$ is updated by reflecting the new $V_{shifted(n)}$ at the $(1-g)$ ratio and maintaining $V_{envelope(n-1)}$ at the g ratio. The g value is calculated by substituting the g_{Attack} calculated in Eq. (1) when $V_{shifted(n)}$ value increases to a larger value than $V_{envelope(n-1)}$ and the $g_{Release}$ calculated in Eq. (2) when decreasing to a smaller value than $V_{envelope(n-1)}$. t_{Attack} and $t_{Release}$ can be used to adjust the sensitivity according to the user's preference. In this paper, t_{Attack} of 5ms and $t_{Release}$ of 10ms are set as default values. Fig. 6 shows an example of $V_{shifted}$ and $V_{envelope}$ as a result of applying the above formula.

Setting a threshold through a calibration process

Using $V_{envelope}$, we determine the onset and offset of the note. The criterion is the threshold set by analyzing the intensity of vibration. There is an onset threshold (Th_{onset}) and offset threshold (Th_{offset}). When the value of $V_{envelope}$ exceeds Th_{onset}, it is judged to be the onset time. When it falls below the Th_{offset} value, it is judged to be the offset time.

The threshold needs to be set differently for each string and may be adjusted depending on the user's sensitivity preference or the change of string tension due to the environment (e.g., temperature, humidity). Therefore, we provide a calibration feature for each string.

At the beginning of calibration, the embedded system will instruct the user not to touch the strings for 2 seconds. During this time, an average value of the input analog data is derived and set as $V_{baseline}$. When $V_{baseline}$ setting is completed, the embedded system guides the user to pluck the six strings one by one. If the user plucks the strings according to the instructions, the vibration values of the six strings are input into the MCU. The MCU analyzes the input vibration values from each string. The vibration caused by one string may influence to the other five sensors due to the resonance phenomenon, vibration transmission, and so on. Therefore, We use I_{max}, which is maximum value of the influence on the envelope due to plucking other strings. For example, the I_{max} of string 1 is the maximum value of the influence when strings 2 to 6 are plucked. This allows the error of certain strings being mistakenly sensed when playing another string to be avoided.

Th_{onset} is set to 1.2 times I_{max}. Th_{offset} is set to an average value of $V_{envelope}$ for 200ms from time when $V_{envelope}$ becomes smaller than $\sqrt{I_{max}}$. If $V_{envelope}$ value of a particular string is equal to or greater than Th_{onset}, we determine that the string has been plucked, and when the $V_{envelope}$ value of the string is less than Th_{offset}, we determine that the note of the string is no longer playing.

EVALUATION

In this section, we evaluate the performance of the guitar that we proposed in terms of the sensing success rate and the error in duration of the note. The performance for the evaluation is performed by the experimenter who is an intermediate guitar player. We evaluate the success rate of multipitch detection for seven basic guitar chords shown in Fig. 7. The experimenter plucks each chord 10 times with various strengths. The plucking strengths are varied from the minimum for generating an audible pitch to the maximum strength of the experimenter. We use log data generated in our embedded system to check the result. The log data include times and sets of a fret number and a string number. The performance is also recorded with a microphone to find a ground truth for the duration of a note.

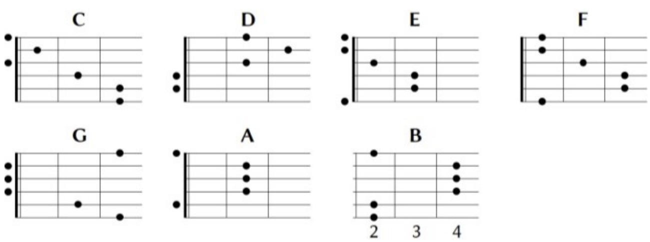


Figure 7. The seven guitar chords used to measure a sensing success rate

The sensing success rate is evaluated as passing or failing regarding whether the user's performance is sensed. The sensing success rate is determined as a "pass" if the performance is sensed well. On the other hand, if it is not sensed correctly, sensed many times, or sensed from other strings, then it is determined as a "fail." This sensing success rate is calculated as in Eq. (4).

$$Sensing\ success\ rate = \frac{The\ Number\ of\ Passes}{The\ Number\ of\ Attempts} \times 100(\%) \qquad (4)$$

A sensing success rate has been measured as 100% in this experiment. Note that if a user plucks guitar strings too weakly, making almost no sound, it cannot be sensed accurately. The sensing success rate may also be affected when the tension of strings changes as time goes, or due to temperature and humidity. In these cases, the user should reset Th_{onset} and Th_{offset} through a calibration process for accurate sensing.

The frequency of calibration depends on the user's preferred sensing accuracy and the degree of change in the guitar strings' state. Normally, the user should recalibrate when the user is tuning the guitar, such as if the user installs new guitar strings or adjusts the tuning of the strings. The guitar will provide constant accuracy, even without recalibration for several weeks, if the tensions of the guitar strings remain constant.

The error in duration of the note is evaluated by checking the difference between the duration recognized by our guitar and the duration found in the recorded audio file. The amplitude scale of audio goes from positive to negative repeatedly within the range between -1.0 and +1.0. We use absolute value of the amplitude ($|amp|$) for defining onset and offset times. When $|amp|$ of the recoded audio is more than 0.1 (white noise) and maintained for 2ms, we consider it the onset of the note. When $|amp|$ is below 0.1 and maintained for 10ms, we define it as the offset of the note. The duration between the onset time and the offset time from the recorded audio is used a ground truth.

The range of the errors in duration of the note for each string is shown in Fig. 8. In this evaluation, the experimenter plucks each string 10 times at various strengths and then mutes after a random time between about 250ms and 850ms. The duration of most common notes (e.g., a quarter note, an eighth note) belongs to this range. The error value in Fig. 8 is the absolute value of the original error value. The average error for all strings is 15.6ms and in most cases the errors are within 40ms. These errors will have to be improved to satisfy instrumentalists and electronic musicians in terms of the perceived quality for action-to-sound latency to be maintained less than 10ms [9]. However, such errors can be ignored in a music transcription application where we map a recognized duration of a note into one of the known durations for common notes. In order words, it is less likely to judge a note as a wrong note because the duration difference between two different notes is sufficiently large. For example, in moderato (bpm=90), a quarter note, an eighth note, and sixteenth note are 666.6ms, 333.3ms, and 166.6ms, respectively.

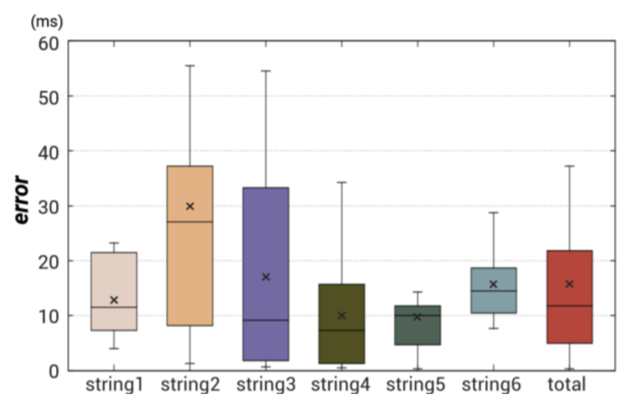


Figure 8. The range of errors in duration of the note in each string

APPLICATIONS

In this section, we introduce various utilization methods of the proposed guitar. A typical application creates musical scores. Using our guitar, musical scores can be generated with performance information by using the results of the pitches and durations of notes detected. After the performance information is analyzed by our scheme, postprocessing is required to make the sensed data a musical score. Since people cannot play with a perfect beat when performing, the duration of the actually played note is mapped to the closest typical note duration. For example, if a user sets the bpm to 90 and plays a note for 700ms, it is determined as a quarter note. Note that the duration of a quarter note with a bpm of 90 is 666.7ms and a dotted quarter note is 1,000ms, so the duration of 700ms should be categorized as a quarter note.

Even if a user plays the same song, created musical scores may differ depending on the style of the user's performance. If a user plays four bars of Mozart's "Twinkle, Twinkle, Little Star" and plays each note similar to the duration of the original notes of the musical score, the generated musical score will be like that of Fig. 9(a). However, if the user plays a note for a shorter duration, an eighth note and a rest may be inserted for a quarter note, as shown in Fig. 9(b). Depending on the implementation of the application, it can also be judged as staccato.

Figure 9. Result of musical score creation depending on performance style

As an extension to the application, after creating one's own musical scores, the user can also share them on the Web using a sheet music sharing platform [12]. Shared musical scores can be downloaded through a smartphone application, and the user can receive LED guidance. In addition, if a user plays the guitar in connection with our smartphone application, TAB score is displayed on the smartphone application in real time, as shown in Fig. 10.

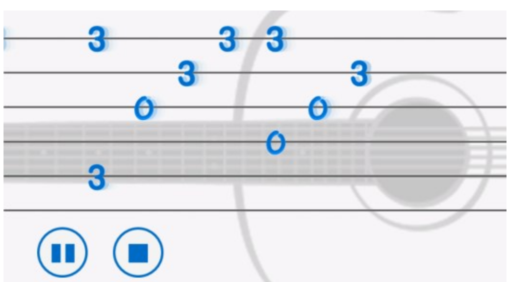
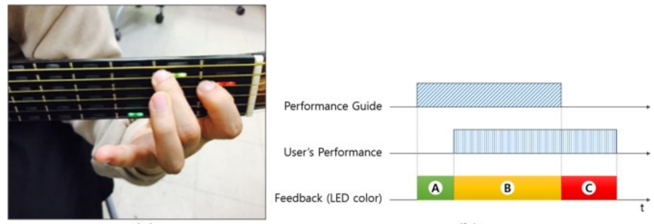


Figure 10. TAB score displayed on a smartphone

Another very useful application is a feedback system. The proposed guitar can analyze the sensed data to determine whether the user is performing correctly and provide feedback in real time. Performance feedback is provided through the LEDs on the guitar's fingerboard, where the fingering position that user has to adopt is illuminated in green. The fingering position LED lights turn yellow if the user performs correctly, while they turn red when the user performs incorrectly. The performance feedback with LED color changes and the feedback scheme for the user's performance result are shown in Fig. 11.



(a) (b)

Figure 11. Feature of real-time feedback: (a) performance feedback by changing LED color, (b) feedback scheme by LED color change

In the case of playing the guitar in connection with the smartphone application, real-time performance feedback is provided through the smartphone, as shown in Fig. 12. Through an interface like a rhythm game, the application displays information about the strings and frets to be fingered, and since the user can check the accuracy and combo scores of their performance, they can gain not only skill improvement, but also enjoyment. Such interactive lessons help guitar learners learn more effectively. In addition, it is expected that various applications will be possible by using sensed data. A video clip that introduces the various features provided by our proposed guitar can be found in [11]

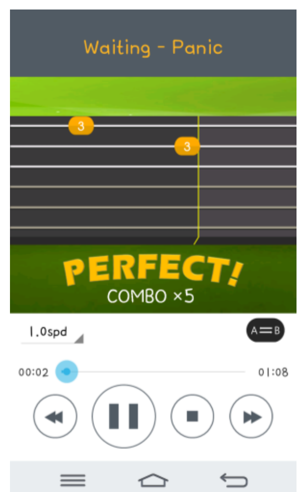


Figure 12. Feature of real-time performance feedback with smartphone

CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a sensor-based guitar that consists of two groups of sensors. The proposed guitar employs a sensing circuit in the fingerboard to recognize the fingering positions of the fretting hand, and includes attached piezo sensors under the saddles to detect the guitar string that are played by picking hand.

Many studies have employed theoretical approaches to recognize musical instrument performance information by analyzing multimedia data such as video and audio.

However, these approaches have to process large amounts of data for analysis which is difficult for a resource constrained embedded system to process in real-time. Therefore, there are still limitations to apply them to real guitars. We have reduced the amount and type of data to be analyzed, with which the existing approaches still could not get sufficient accuracy. Instead, we have designed appropriate sensor circuits. Fingering positions and plucked string judgments for multipitch estimation were made recognizable by hardware circuitry. This was yet a challenging task for an embedded system with limited number of input/output pins. The analog data from the piezo sensor was analyzed together with digital data from the hardware circuitry to precisely calculate the onset time, offset time, and duration of the note. This is one of our main contributions in this paper. Other contributions for the interactive learning environment and tactile interface can be found in our developed features or applications such as music transcription, a sheet music sharing platform, an LED feedback guitar, and a smartphone rhythm game.

In the future, one may study how to utilize only one of two sensor groups or re-design a hardware circuitry to get partial information, for example, the fret number without knowing the string number. This may improve the accuracy of existing audio-based analysis with a relatively simplified hardware configuration. We also plan to evaluate user satisfaction with our guitar.

ACKNOWLEDGMENTS

This work (Grants No.C0238987) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2015.

REFERENCES

1. Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. 2005. A Tutorial on Onset Detection in Musical Signals. IEEE Transactions on Speech and Audio Processing. 13, 5: 1035-1047. http://dx.doi.org/10.1109/TSA.2005.851998
2. Frédéric Bevilacqua, Nicolas Rasamimanana, Emmanuel Fléty, Serge Lemounton, and Florence Baschet. 2006. The augmented violin project: research, composition and performance report. In *Proceedings of the international conference on New Interfaces for Musical Expression* (NIME '06), 402-406. http://nime.org/proceedings/2006/nime2006_402.pdf
3. Giovanni Costantini, Massimiliano Todisco, and Giovanni Saggio. 2011. A sensor interface based on sparse NMF for piano musical transcription. In *Advances in Sensors and Interfaces (IWASI)*, 157-161. https://doi.org/10.1109/IWASI.2011.6004708
4. Fretlight. 2017. Wireless Guitar. (Dec 2017). http://fretlight.com/collections/guitars
5. Fret Zealot. 2017. Fret Zealot. (Dec 2017). http://fretzealot.com/
6. Tobias Grosshauser, and Gerhard Tröster. 2013. Finger Position and Pressure Sensing Techniques for String and Keyboard Instruments. In *Proceedings of the international conference on New Interfaces for Musical Expression* (NIME '13), 479-484. http://nime.org/proceedings/2013/nime2013_286.pdf
7. Tobias Grosshauser, and Gerhard Tröster. 2013. Optical bow position, speed and contact point detection. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* (UbiComp '13), 211-214. https://doi.org/10.1145/2494091.2494159
8. Tobias Grosshauser, and Gerhard Tröster. 2014. Musical instrument interaction: development of a sensor fingerboard for string instruments. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction* (TEI '14), 177-180. https://doi.org/10.1145/2540930.2540956
9. Robert H. Jack, Tony Stockman, and Andrew McPherson. 2016. Effect of latency on performer interaction and subjective quality assessment of a digital musical instrument. In *proceeding of the Audio Mostly* (AM '16), 116-123. http://dx.doi.org/10.1145/2986416.2986428
10. Andrew McPherson. 2012. TouchKeys: Capacitive Multi-Touch Sensing on a Physical Keyboard. In *proceedings of the 12th international conference on New Interfaces for Musical Expression* (NIME '12), 195-198. http://nime.org/proceedings/2012/nime2012_195.pdf
11. NSERLAB. 2017. Real-time Recognition of Guitar Performance Using Two Sensor Groups for Interactive Lesson. Video. (November 28, 2017). Retrieved November 28, 2017 from https://youtu.be/dpJqdGgdPVc
12. Laurel S. Pardue, Christopher Harte, and Andrew P. McPherson. 2015. A Low-Cost Real-Time Tracking System for Violin. Journal of New Music Research. 44, 4: 305-323. http://dx.doi.org/10.1080/09298215.2015.1087575
13. Soonuk Seol, Yejin Shin, and Kangwon Lee. 2016. Learning Guitar with an Embedded System. Contemporary Engineering Sciences. 9, 12: 553-560. http://dx.doi.org/10.12988/ces.2016.6441
14. Fabian-Robert Stöter, Michael Müller, and Bernd Edler. 2015. Multi-Sensor Cello Recordings for Instantaneous Frequency Estimation. In *Proceedings of the 23rd ACM international conference on Multimedia* (MM '15), 995-998. https://doi.org/10.1145/2733373.2806384

15. Nick Summers. 2014. gTar: This iPhone-powered guitar isn't the educational instrument you've been dreaming of. (January 25, 2014). Retrieved August 2, 2017 from http://tnw.to/q3Fnh

16. Tim Vets, Jonas Degrave, Luc Nijs, Federica Bressan, and Marc Leman. 2017. PLXTRM: Prediction-Led eXtended-guitar Tool for Real-time Music applications and live performance. Journal of New Music Research. 46, 2: 187-200. https://doi.org/10.1080/09298215.2017.1288747

17. Kazuki Yazawa, Daichi Sakaue, Kohei Nagira, Katsutoshi Itoyama, and Hiroshi G. Okuno. 2013. Audio-based guitar tablature transcription using multipitch analysis and playability constraints. In *Acoustics, Speech and Signal Processing* (ICASSP '13), 196-200. https://doi.org/10.1109/ICASSP.2013.6637636

18. Jun Yin, Ye Wang, and David Hsu. 2005. Digital violin tutor: An integrated system for beginning violin learners. In *proceedings of the 13th annual ACM international conference on Multimedia*, 976-985. http://dx.doi.org/10.1145/1101149.1101353

19. Bingjun Zhang, Jia Zhu, Ye Wang, and Wee Kheng Leow. 2007. Visual analysis of fingering for pedagogical violin transcription. In *Proceedings of the 15th ACM international conference on Multimedia* (MM '07), 521-524. https://doi.org/10.1145/1291233.1291361