Support System for Improvisational Ensemble Based on Long Short-Term Memory Using Smartphone Sensor

Haruya Takase
Shun Shiramatsu
Graduate School of Engineering, Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan
h.takase@srmtlab.org
siramatu@nitech.ac.jp

ABSTRACT

Our goal is to develop an improvisational ensemble support system for music beginners who do not have knowledge of chord progressions and do not have enough experience of playing an instrument. We hypothesized that a music beginner cannot determine tonal pitches of melody over a particular chord but can use body movements to specify the pitch contour (i.e., melodic outline) and the attack timings (i.e., rhythm). We aim to realize a performance interface for supporting expressing intuitive pitch contour and attack timings using body motion and outputting harmonious pitches over the chord progression of the background music. Since the intended users of this system are not limited to people with music experience, we plan to develop a system that uses Android smartphones, which many people have. Our system consists of three modules: a module for specifying attack timing using smartphone sensors, module for estimating the vertical movement of the smartphone using smartphone sensors, and module for estimating the sound height using smartphone vertical movement and background chord progression. Each estimation module is developed using long short-term memory (LSTM), which is often used to estimate time series data. We conduct evaluation experiments for each module. As a result, the attack timing estimation had zero misjudgments, and the mean error time of the estimated attack timing was smaller than the sensor-acquisition interval. The accuracy of the vertical motion estimation was 64%, and that of the pitch estimation was 7.6%. The results indicate that the attack timing is accurate enough, but the vertical motion estimation and the pitch estimation need to be improved for actual use.

Author Keywords

Body motion, improvisational ensemble, smartphone sensor, LSTM, tonality

CCS Concepts

- **Applied computing** → Sound and music computing
- **Human-centered computing** → Usability testing

1. INTRODUCTION

It is difficult for inexperienced people who do not know aspects of music theory such as chord progression and have no experience playing instruments to participate in musical interaction by emitting harmonious sounds that do not compromise their sense of tonality. Music events and artist concerts held for the promotion of the local community are examples of inexperienced people actively participating in music interaction. Traditionally, the means of participation in such performances by inexperienced people have been limited to clapping, shouting, and moving their bodies to the rhythm. We aim to develop a system for enabling inexperienced people to engage in musical activities together with those with music experience.

Hatano [1] identified three processing aspects in the development of melodic singing ability and cognitive processing of melodic listening: (1) rhythm, (2) pitch contour, and (3) tonality, as shown in Figure 1. The melody line in (2) refers to the movement pattern of the pitch rising and falling, and it is relatively easy for beginners to roughly understand the rhythm and whether a note has risen or fallen. The tonalities in (3) are explained in terms of central notes, major/minor notes, scales, chords, etc. Understanding the tonalities is one of the biggest obstacles for music beginners. Therefore, tonality is a reason inexperienced musicians have a sense of discomfort when trying to perform musical activities. We hypothesized that a music beginner cannot determine tonal pitches of melody over a particular chord but can use body movements to specify the pitch contour (i.e., melodic outline) and the attack timings (i.e., rhythm). That is, we assume that an inexperienced user can specify the pitch contour by her/his vertical motion of hands without knowledge of chords or tonality. If the lack of users' knowledge of tonality is addressed by some computational support, inexperienced people can also participate in more fulfilling musical activities.

Suga [2] argues that physical movement, such as tracing a melody, is an effective way to promote musical understanding. Therefore, it is appropriate to use body movements as input for the melody line. In addition, since most people usually carry a smartphone, it can be used in urgent situations without being constrained by location.

In a previous study, Mizuno et al. [3, 4] have developed an improvisational ensemble support system using a smartphone sensor. In this study, we implemented a simple position tracking method that traces the vertical movement of the smartphone from the values measured by sensors (accelerometer, gyro sensor, etc.) mounted on the smartphone. Therefore, Mizuno et al. proposed a method for estimating the user's movement and the pitch to be output from the traced movement data as training data using Bayesian network stochastic model estimation. The estimation is divided into three components: (1) attack timing estimation, (2) up-and-down pitch estimation, and (3) sound name estimation. The estimation accuracy of each element of the test data was (1) F value = 0.37, (2) correct answer rate = 56%, and (3) correct answer rate = 45%. The results of the subject experiment were (1) rating = 2.38, (2) rating = 1.75, and (3) rating = 4.50 on a 7-point scale (7 being the highest, meaning satisfactory for the users). The Bayesian network in the previous study raised problems such as estimation of attack timing outside the normal timing and estimation of the up-and-down movement of the pitch contrary to the user's intended pitch movement, so using a different model was considered in this study.

There are many studies using long short-term memory (LSTM) for melody generation. Kuhara et al. [5] used LSTM as an input to assist users with no musical knowledge in creating music, using the features of existing songs and user-created melodies as inputs. A system for estimating and presenting candidate melodies is implemented. Unlike previous research, the system does not receive a melody but rather generates a suitable melody for the BGM by receiving the smartphone's sensor values and BGM to help the user perform.

Based on these studies, we aim to improve the accuracy of the conventional research system by changing the format of the training data and using LSTM for the estimation model.

NIME'20, July 21-25, 2020, Royal Birmingham Conservatoire, Birmingham City University, Birmingham, United Kingdom.

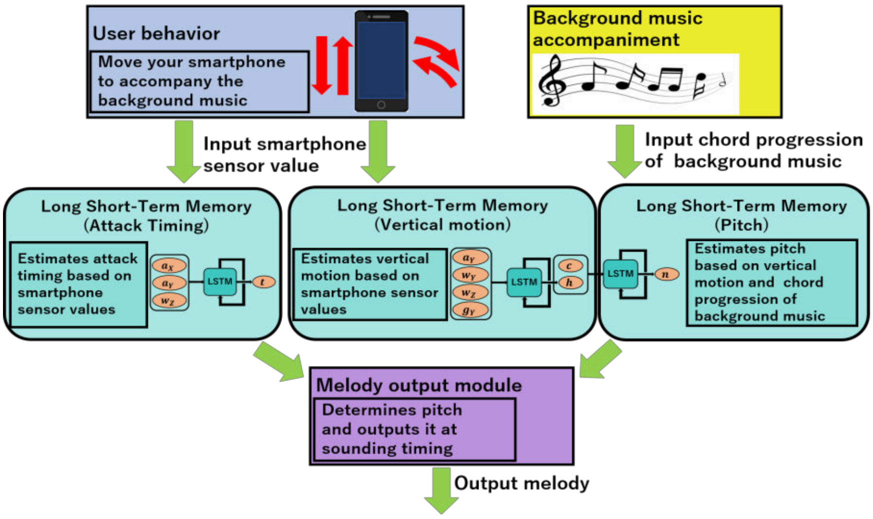
User behavior
Move your smartphone to accompany the background music
Background music accompaniment
Input smartphone sensor value
Input chord progression of background music
Long Short-Term Memory (Attack Timing)
Estimates attack timing based on smartphone sensor values
a_X
a_Y
w_Z
LSTM
t
Long Short-Term Memory (Vertical motion)
Estimates vertical motion based on smartphone sensor values
a_Y
w_Y
w_Z
g_Y
LSTM
c
h
Long Short-Term Memory (Pitch)
LSTM
n
Estimates pitch based on vertical motion and chord progression of background music
Melody output module
Determines pitch and outputs it at sounding timing
Output melody

Figure 1. System architecture

2. SYSTEM ARCHITECTURE

Figure 1 shows our system architecture. First, the user holds the smartphone vertically, moves the smartphone up and down to accompany the background music, and shakes the smartphone according to the background music, as shown in Figure 2. By using the values from the smartphone sensors based on the user's motion as input values, the attack timing estimation module and the vertical movement estimation module estimate the attack timing and vertical movement, respectively. Next, the pitch estimation module estimates the pitch so that it is not dissonant based on the background music and the user's motion using the estimated vertical movement and the chord progression information of the background music as input values. Finally, the melody output module outputs the estimated pitch at the estimated attack timing. Currently, the output sound is assumed to consist of a single sound and a decay sound.

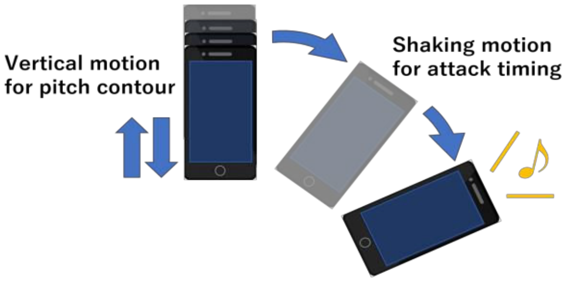

Figure 2. How to use system.

3. DESIGNATION METHOD OF ATTACK TIMING AND PITCH CONTOUR

3.1 Motion sensor mounted on smartphone

The rhythm and melody are determined from the user's body movements by using the motion sensor mounted on the smartphone, and these sensor values are used as input values. The sampling rate is *200* Hz (*1* frame = about *5.04* milliseconds). The axis of the smartphone sensor is shown in Figure 3. Although the acceleration, gravitational acceleration in each axis, angular velocity, and rotation about each axis can be obtained, verifying the accuracy of all feature combinations would take a great deal of time, so, in this study, we obtain the sensor values in the axial direction and around the axis that will be greatly affected by the swing motion of the smartphone.

Specifically, the features are acceleration a_X in the *X-axis* direction, acceleration a_Y in the *Y-axis* direction, rotation r_Z about the *Z-axis*, angular velocity w_Y about the *Y-axis*, angular velocity w_Z about the *Z-axis*, and gravitational acceleration g_Y in the *Y-axis* direction. From these features, the most accurate combination is adopted as input based on the results of the evaluation experiments on the test data.

3.2 Kinect 3D camera sensor

A Kinect 3D camera sensor is used to obtain the teacher data in the training of the up-and-down estimation model. The sampling rate is *30* Hz (*1* frame = about *34* milliseconds).

The teacher data is specifically the height of the thumb of the hand holding the smartphone. We define the thumb height value as the relative *Y-axis* coordinates of each frame when the *Y-coordinate* on the window of the thumb of the user holding the smartphone at a distance of *3* meters from the Kinect 3D camera is set to *0* at the time of the sensor value acquisition.

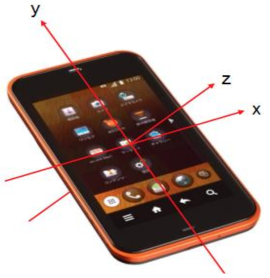

Figure 3. Axis of smartphone sensor

3.3 Collecting training data

We need to collect the training data of the relationship between the smartphone sensor values and pitch contour to develop the LSTM model.

When the user moved the smartphone up and down according to the pitch and rhythm of the melody prepared in advance, the smartphone sensor value was recorded about every *5.04* milliseconds (*1* frame). At the same time, the height information of the user's hand recognized by the Kinect 3D camera was recorded about every *34* milliseconds. The recording start time of the sensor value of the smartphone and that of the Kinect 3D camera are synchronized by Bluetooth.

Also, the prepared melody was made with a MIDI sound source, and the attack timing and pitch were recorded in milliseconds. We used 50 MIDI files from the RWC popular music database to obtain the dataset for training and verifying the accuracy of the pitch estimation model and converted the main melody from the MIDI files into a CSV file with note number and time (in seconds). For the input chord progressions, we used the annotation data of songs from the AIST Annotation for the RWC Music Database, from which the main melody data was obtained. The chord names and times (in seconds) were downloaded as a LAB file and used, and the data of 50 songs gave a total of 18,867 notes in terms of the number of main melody notes.

The MIDI sound source used to acquire the dataset of the attack timing estimation model and learn and verify the accuracy of the vertical motion estimation model consists of {C4,D4,E4,F4,F4,G4,A4,B4,C5}, which are randomly arranged four times at equal intervals of bpm80 and bpm40, respectively. In this study, data was collected 12 times at each BPM for a total of 768 notes.

The collected data were simply combined in the time direction, and the combined data were divided into 60% training data, 20% validation data, and 20% test data for training and accuracy verification of the model.

4. IMPLEMENTATION

In this study, we divide the output of the performance sound into the following three models and propose a method using these three LSTM models.

- Model 1: Attack timing estimation model
- Model 2: Vertical motion estimation model
- Model 3: Pitch estimation model

For the input of the attack timing estimation model and the vertical motion estimation model, the most accurate combination in the test data is adopted from all the combinations of features presented in Section 3.1. The attack timing estimation model and the vertical motion estimation model are trained by the hold-out method because of the large amount of data, and the pitch estimation model is trained by 10-fold cross validation.

4.1 Attack timing estimation model

Figure 4 shows the LSTM model for estimating the attack timing. We adopt $\{a_X, a_Y, w_Z\}$, which were the most accurate input values of this model, as a result of feature selection. The LSTM's timesteps use the past three frames with the highest accuracy.

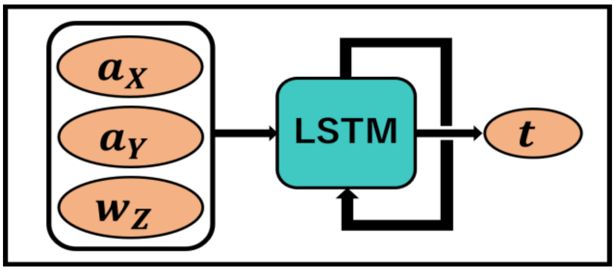

Figure 4. Attack timing estimation model.

As shown in Figure 5, the teacher data is divided into three sections: {swing-down section, stop section, and other motion section}. We defined the attack timing as time t when the swing-down section shifts to the stop section.

To create the correct answer data, it is first necessary to correct the difference between the recording start time of the smartphone sensor and the playback start time of the MIDI sound source. As a result of the verification, the recording start time was found to be 200 [ms] earlier on average than the sound source playback start time, so 200 [ms] was manually subtracted from the recording time of the smartphone sensor.

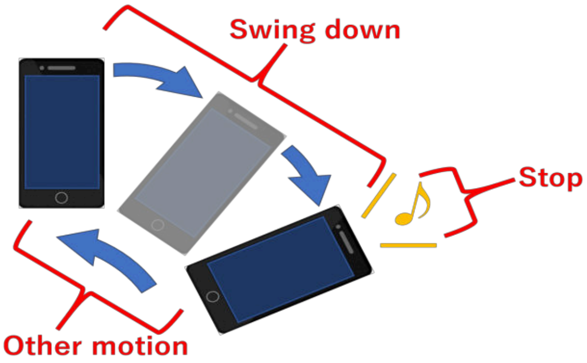

Figure 5. Three sections of attack timing estimation model.

Next, it is necessary to correct the time lag between the attack timing that the user perceives and the attack timing of the actual MIDI sound source. As shown in Figure 6, we define the starting point of stop as time t_s when $a_X(\mathrm{t_s})$ has the maximum values between before and after a certain time from the attack timing of the MIDI sound source. We define the starting point of swing-down as time t_d when $r_z(t_d)$ has the minimum values between before and after a certain time from the attack timing of the MIDI sound source. We define the starting point of other motion as time t_o when $r_z(t_o)$ satisfies $r_z\ (t_o) <= \mathrm{r}_z\ (t_s) - 0.02$ and t_o satisfies $t_o > t_s + 25$ frames . The thresholds are empirically determined so that all the starting points in the data are obtained. Therefore, it is possible that this threshold may be changed when new data are collected to create training data.

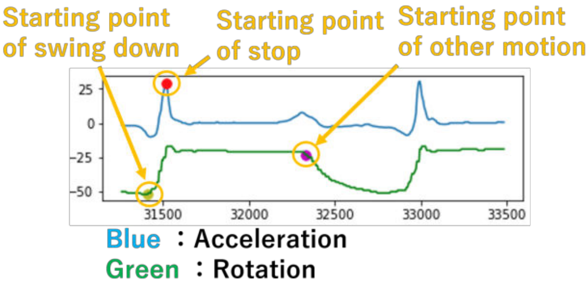

Figure 6. How to calculate the starting point for each section.

Finally, we defined the swing-down section as the section from the starting point of swing-down to the starting point of stop. Similarly, we defined the stop section as the section from the starting point of stop to the starting point of another motion and defined the other motion section as the section from the starting point of the other motion to the starting point of swing-down. We trained the model to classify these three sections.

4.2 Vertical motion estimation model

Figure 7 shows the LSTM model for estimating vertical movement. We adopt $\{a_Y, w_Y, w_Z, g_Y\}$ as the input values of this model as a

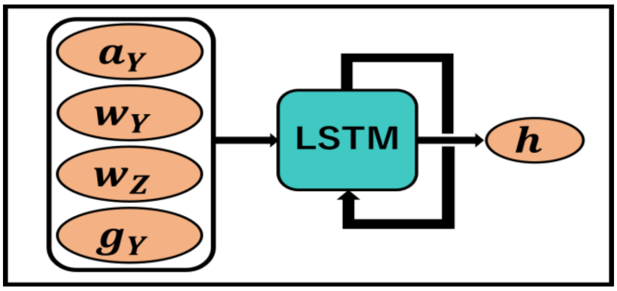

Figure 7. Vertical motion estimation model.

result of selecting the features. The LSTM's timesteps use the past 10 frames with the highest accuracy.

Let s be the summation of h between the attack timings. When s satisfies $s < -2.5$ cm, it is recognized as Down, $h = -1$. When $s > -2.5$ cm and $s < 2.5$ cm, it is No Change, $h = 0$, and when $s > 2.5$ cm, it is Up, $h = 1$, so this model estimates $h = \{-1, 0, 1\}$.

4.3 Pitch estimation model

Figure 8 shows the LSTM model for estimating the output pitch. At the time when the attack timing estimation model is estimated to be the attack timing, n, which represents the pitch to be output, is estimated by using the vertical movement h generated by the model to estimate the vertical movement and c, which represents the chord progression of the background music, as input values. H has three dimensions, as explained in the previous section. C defines the codename C as a one-hot vector. The total number of input values is 239 dimensions, since there are 236 types of chords in the 50 popular music pieces in the RWC music database. In the LSTM time step, the last three frames with the highest accuracy are used.

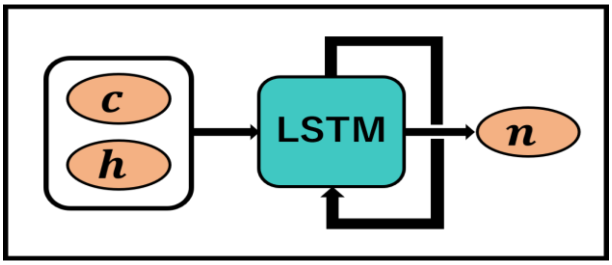

Figure 8. Pitch estimation model.

The output range is defined as the pitch range of the main melody of the training data. Since the main melodies included in the data collected in this study were in the range of MIDI note numbers 54 to 91, this range is the output range of the system.

5. PERFORMANCE EVALUATION

The test data was used to confirm the accuracy of the three LSTM models: (1) attack timing estimation, (2) vertical motion estimation, and (3) pitch estimation.

5.1 Accuracy of attack timing estimation

We verify the accuracy of the attack timing estimation model. The total number of test data frames is 34,755, which includes 103 sound generation timings.

Table 2 shows the accuracy when the features in Figure 3 are used as input values and when the features in Figure 5 are used as input values. If it is estimated that the attack timing is within ± 10 frames from each correct answer timing, only the first attack timing will be correct. If the attack timing is not estimated within 10 frames after each correct timing, it is counted as an estimation error. If the attack timing is estimated 10 frames or more earlier than each correct answer timing or if it is estimated as the second or higher attack timing at each correct answer timing, it is counted as a useless estimate. From these results, it can be said that the accuracy is very high because there is no error in the estimation, and the average absolute error is smaller than 1 frame (about 5.04 milliseconds).

5.2 Accuracy of vertical motion estimation

We verify the accuracy of the vertical motion estimation model. The total number of test data frames is 34,755 frames, which includes 103 sound generation timings.

Table 3 shows the accuracy when the features in Figure 6 are used as input values. The accuracy rate, recall rate, and precision rate between the vertical motion estimation result and the correct vertical motion between each attack timing are shown.

Table 1. Accuracy of attack timing estimation

Feature	$\{a_X, a_Y, w_Z\}$
Timesteps	3
Estimation error	0 times
Useless estimates	0 times
Mean absolute error	2.37 milliseconds

Table 3. Accuracy of vertical motion estimation

Feature	$\{a_Y, w_Y, w_Z, g_Y\}$	
Timesteps	10	
Accuracy	0.64	
	Recall	Precision
Down	0.82 (32/39)	0.76 (32/42)
No Change	0.36 (4/11)	0.17 (4/24)
Up	0.54 (21/39)	0.91 (21/23)

Table 4. Accuracy of pitch estimation

Feature	$\{c, h\}$	
Timesteps	1	
Accuracy	Mean recall	Mean precision
0.076	0.024	0.052

The result was by no means highly accurate. The recall of Up was low because Up was often determined to be No Change. This may be improved by reducing the positive value of the range where No Change is the sum of the relative values of the vertical movement estimated during the attack timing. On the other hand, with respect to Down and Up, a high precision means that unintended behavior is rare when the system is moved.

5.3 Accuracy of pitch estimation

The accuracy of the pitch estimation model is verified. The total number of notes in the test data is 3371, and the accuracy was evaluated by 10-fold cross validation, and in this study, the correct answer was given when the pitch estimated from the test data as input matched perfectly with the pitch of the sound source in the test data. Table 4 shows the correct answer rate, mean reproduction rate, and mean fit rate for the test data.

The results show that the accuracy rate, average recall rate, and average precision rate are all low. In this verification, the accuracy was low because the pitch estimation was correct only when the exact pitch was estimated with respect to the correct pitch. It is permissible in future verifications to regard the pitch estimation as correct when the estimated pitch up-and-down movements match the actual pitch up-and-down movements and when the estimated pitch is included in the available note scale of the chord.

6. CONCLUSION

In this study, we examined a performance interface that predicts the sound that matches the background music by intuitively inputting rhythm and melody lines. To determine the rhythm and melody, the acceleration of the user's motion is measured using the motion sensor mounted on the smartphone of the user. We have proposed an LSTM model for estimating the pitch, attack timing, and vertical movement using LSTM with the measured features as input values.

By having the user actually hold the smartphone and move the smartphone in accordance with the melody of the background music, learning data without a time lag is created from the values measured by the smartphone sensors, and the optimal features and parameters for each LSTM model are created. After verification, the estimation accuracy was verified using test data. As a result of the verification, it became clear that it was possible to specify the attack timing, but it was difficult to estimate the vertical movement and specify the appropriate pitch for the background music.

In the future, we will improve the accuracy of vertical motion estimation using smartphones, introduce and improve the available note scale in pitch estimation, and finally implement these models on smartphones and perform subject experiments. In the future, we aim to implement an ensemble function for multiple users using multiple devices.

ACKNOWLEDGEMENTS

This work was partially supported by JSPS KAKENHI (17K00461).

REFERENCES

[1] Hatano, G. "*Music and Cognition*," Tokyo University of Tokyo Press, 2007 (in Japanese).

[2] Suga, M. "Possibilities of Participatory Music Concerts Incorporating Body Expressions: The Logic of Canon", Wakayama University Faculty of Education Bulletin of the Research Center for Practical Practice 18, pp. 121-129, (2008)

[3] Ichinose, S., Mizuno, S., Shiramatsu, S., and Kitahara, T. "Two Approaches to Supporting Improvisational Ensemble for Music Beginners based on Body Motion Tracking," in *International Journal of Smart Computing and Artificial Intelligence*, 3(1), p. 55-70, 2019.

[4] Mizuno, S., Ichinose, S., Shiramatsu, S., and Kitahara, T., "Support System of Improvisational Ensemble Based on User's Motion Using Smartphone Sensors," in *Proceedings of the 12th International Conference on Knowledge, Information and Creativity Support Systems*, pp. 143-148, Nov. 2017.

[5] Kuhara, S, Ushitani, T. "A Composition Support Method Based on Learning Existing Songs Using LSTM", DEIM Forum 2018, F3-2, (2018)