Runner's Jukebox: A Music Player for Running Using Pace Recognition and Music Mixing

Sanghoon Jun, Jehyeok Rew, Eenjun Hwang
Department of Electrical & Computer Engineering
Korea University
145 Anam-Ro, Seongbuk-Gu, Seoul 136713, Korea
e-mail: ysbhjun@korea.ac.kr, rjh1026@korea.ac.kr, ehwang04@korea.ac.kr

***Abstract*—In this paper, we propose a smartphone music player called Runner's Jukebox (RJ) that is especially effective for walking and running activity. RJ consists of user pace speed recognizer, music archive with feature data, and dual music player. To show its effectiveness, we assume two scenarios in this paper: (i) users are required to catch up with music tempos of predefined playlist. (ii) Songs are played dynamically or changed to match with the speed of user pace. We present novel pace recognition algorithm that enables RJ to recognize user pace in any orientations and even in the pocket. Various methods for playing and selecting songs based on user pace speed are presented. In addition, in order for seamless playback of songs, two mixing schemes are presented for music player. To evaluate the performance of our proposed scheme, we carried out several experiments on an Android platform. We report some of the results.**

Keywords- Interactive music; Mobile application interface; Pace tracking; Context-aware computing; Wearable computing.

I. INTRODUCTION

Music can be used as a good stimulus for increasing effects of physical exercise [1]-[3]. It helps exercisers to have positive mood while workout and hence leads to better performance. In other words, exerciser can use music as a means to improve their exercise workout.

Explosively growing popularity of smartphone and wearable devices has brought entirely new interfaces and applications to human beings. Many studies have been done recently based on these state-of-the-art technologies. For instance, in the traditional music streaming services, users play songs by just clicking or selecting the desired ones. Hence, they should decide the playback list beforehand or continuously, which makes them easily fed up with the tasks or unsatisfied with the outcome.

In this paper, we present a smartphone music player called Runner's Jukebox (RJ), which takes different approaches to playing music compared to the traditional one. By combining the sensor data analysis of smartphone (or wearable devices) and music feature extraction, the music player can play songs to be matched with user pace and adjust the playback speed dynamically to catch up with the pace afterwards. The idea of music player tracking user pace is not novel in itself. A few studies have been done for tracking and synchronizing a song to user's pace [4-8]. The contribution of this paper focuses on the practical implementation of music player integrating aforementioned technologies on Android smartphone and wearable devices and some evaluations of actual user behaviors. The rest of this paper is organized as follows: In Section 2, we present a brief overview of the related works. Section 3 presents the overall system architecture, music play modes and song mixing method. Section 4 describes the experiments that we performed and some of the results, and the last section concludes the paper.

II. RELATED WORKS

Wijnalda et al. [6] proposed the personalized music system called IM4Sport. It helps select music that suits a training program, changes playback to reflect or guide current sport performance, and collects data for adapting training programs and music selections. They proposed the prototype system that consists of personal computer, a portable music player, a heart sensor strap and a pedometer. Moens et al. [7] presented D-Jogger, a music interface that makes use of body movement to dynamically select music and adapt its tempo to the user's pace. While implementing the interface, they focused on the entrainment which is the synchronization of music and walking. Oliver et al. [8] presented MPTrain, a mobile phone based system that helps users to achieve their exercise goals. Using extra physiological sensors of mobile hardware, the proposed system allows the user to select desired exercise pattern such as heart-rate goal. While several studies are based on their own devices and systems, we focus on the development of algorithms on popular and general mobile platforms such as android platform without any additional devices

A few applications are already available for this purpose. Recently, applications such as CruiseControl, TempoRun and TrailMix are released to play and approximately match a song to user pace. In the case of RockMyRun, mixed tracks with certain tempo are provided to the user for better efficiency of running or walking. However, it is designed to provide pre-mixed track without any customization or dynamic pace matching. In this paper, we focus on more advanced features such as mixing songs continuously and adaptively based on accurate user pace recognition to

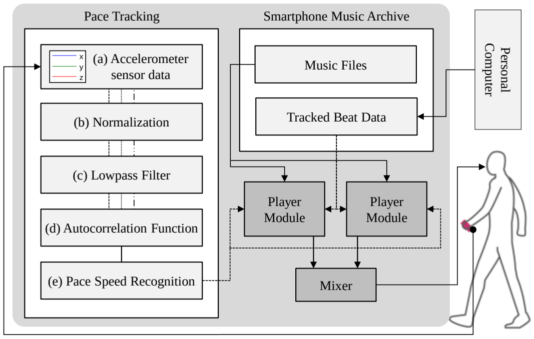

Figure 1. Overall system architecture

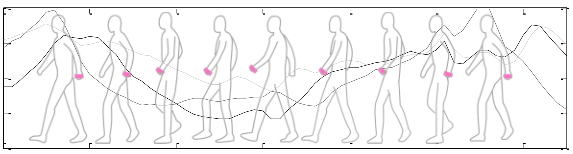

Figure 2. 3-axis acceleration changes of human pace

provide better exercise efficiency and convenience for joggers.

III. RUNNER'S JUKEBOX

A. Implementation

The RJ application consists of three main components: (1) Pace recognizer based on device sensor data. (2) Music archive that contains music features such as beat and tempo. (3) Two music players and one mixer that support diverse playing modes and music mixing. Its overall architecture is shown in Fig. 1.

B. Pace Recognizer

In RJ, we suppose that the user holds the smartphone on his hand or armband while the user walks or runs. Since the smartphone gets acceleration while swinging his/her arm, the pace can be detected by analyzing acceleration sensor data of smartphone. Fig. 2 shows an example of acceleration changes of human pace. Acceleration data may change unpredictably while he/she walks or runs.

The Android platform provides hardware-based sensor data for monitoring the motion of a device. For detecting the pace, the accelerometer sensors in x, y and z axis are used. The platform can monitor sensor data with short delays (0ms to 60ms).

In this paper, we propose an algorithm for recognizing the pace. Fig. 3 shows the detailed steps for pace recognition. First, 3-axis acceleration sensor data in sample window is acquired from device. In order to remove gravity or unnecessary acceleration, three dimensional data are normalized to zero-center, respectively. Similarly to the method presented in [7], low-pass filter based on Butterworth IIR filter is applied to remove noises. In order to observe the periodicity of user pace, an auto-correlation function is computed from each axis sensor data. Three auto-

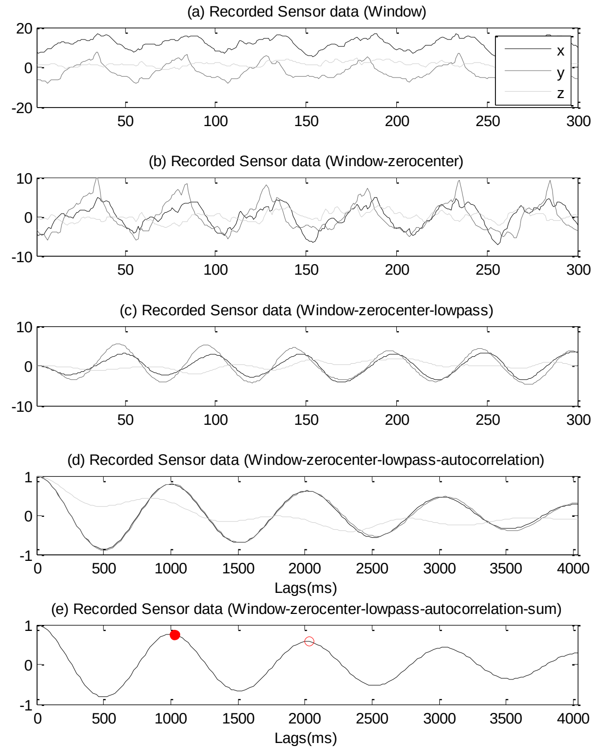

Figure 3. Recognizing pace from acceleration sensor data

correlation functions (ACF) are fused together using the following equation to calculate their ACFF:

$$ACFF(\tau) = \sum_{d \in \{x,y,z\}} \mathrm{var}(d) \times ACF_d(\tau) \quad (1)$$

where var(d) is the variance of d-axis sensor data within a window. Among the three axis sensor data, we consider the axis with the largest variance has more primary information for recognizing periodicity. This approach makes the pace recognition independent of any orientation of the sensors. This is important since user's gripping style and orientations of wearable device can be different.

In order to represent the speed of user pace quantitatively, we define SWing Per Minute (SWPM).The period of swings can be observed by finding the lag having the first local maximum value of auto-correlation function. The following equation shows how to calculate the SWPM from the period.

$$SWPM = \frac{60}{FirstPeakLag(ACFF)} \quad (2)$$

The value of SWPM indicates how fast the user walks or runs. For instance, we empirically found that the SWPM of typical walking adults is between 10 and 40. The detailed experiment for SWPM values will be described in later section.

Typical time taken for a human to swing his arm is less than 3 seconds. Therefore, we calculated the SWPM using 3 seconds of window in every 20 milliseconds. User can speed

 ISBN: 978-1-61208-400-8

up or down instantly for synchronizing his/her pace to the playing song. To make the SWPM closer to user speed, we used the moving average window method.

C. Music Archive Construction

In order to synchronize song to user pace or play music continuously, beats within a song should be tracked. In this paper, we employed BeatRoot, the java based beat tracking system that proposed in [9]. Beats are tracked and their timestamps are stored as a text file in a personal computer as shown in Fig. 1. Subsequently, tempo in terms of beats per minutes (BPM) is calculated by the average intervals of beats.

D. Music Player

While songs are played in RJ, their tempo needs to be adjusted according to the user's SWPM. Since the SWPM is correlated to an arm swing of two footsteps, the proper tempo of song is calculated by the following equation

$$BPM = 2 \times SWPM \tag{3}$$

SoundTouch, which is a sound processing library implemented on C++ platform, is employed for adjusting playback speed without pitch variation [10]. Java native interface is used to enable the library on java platform. Using the library, songs can be played with its playback speed adjusted dynamically according to user SWPM.

The RJ player provides two modes for music selection and two options in each mode for adapting its playback to the user pace.

1) User Mode

In this mode, the player plays songs in the playlist which was made by the user beforehand. This mode provides two particular options: fixed and dynamic pace. With the fixed pace option, tempo of all songs in the playlist is changed to some fixed BPM. On the other hand, with the dynamic pace option, the songs in the playlist are played sequentially and its playback speed is changed dynamically depending on user SWPM.

2) System Mode

In this mode, the playlist is made automatically by the system and two options are supported: Beat-aware and program. With the beat-aware option, the system monitors user SWPM continuously. When any change is detected in the user SWPM, the player finds songs having the most similar tempo. In other words, if the user changes his/her speed up or down beyond the adjustable tempo range of current music, the player stops current music and plays another song with corresponding tempo. Another option is program where songs are played based on the predefined programs. For instance, if the user defines a program that consists of 5 minute run and 5 minute walk with 5 repeats, the system automatically prepares a playlist of songs appropriate for the program.

3) Song Mixing

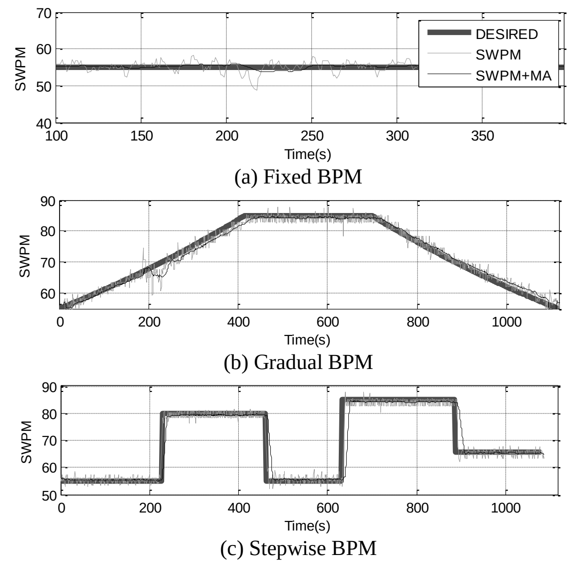

Figure 4. Examples of pace matching to music tempo

RJ has two player modules for music mixing to support seamless play. In other words, they play songs seamlessly in such a way that songs are sounded like one. The two player modules synchronize their music play based on the beat feature in the music archive. There are two options for song mixing: crossfading and cutting. In the crossfading mix, two songs are mixed gradually by volume transition over time. This method can be used when the SWPM is steady. In the cutting mix, a new song starts as soon as current song ends. This method is used for mixing music when the SWPM changes suddenly.

IV. Evaluation

To evaluate the performance of our scheme, we performed several experiments for 10 females and 10 males. 15 of them are 20s and the others are 30s. Also, 5 of them are daily exerciser and 12 of them exercise regularly. The others exercise rarely. We collected 100 songs (MP3 files) of various genres from a song streaming service, which include electronic, hip-hop, pop and rock. Since our evaluation is related to exercise, we collected the user's custom playlists that were programmed for their exercise or workout.

A. SWPM Ranges

Based on the proposed pace metric (SWPM), we collected the actual walking/running records from 20 subjects. The subjects were asked to walk or run about 2km with their own pace. Average SWPMs of walking and running are 52.6 and 82.9, respectively. We implemented a prototype Runner's jukebox and tested it on two android smartphones: LG Optimus G (Snapdragon S4 Pro 1.5GHz Quad-core processor and Android OS v4.1) and Samsung Google Nexus S (Cortex-A8 1GHz processor Android OS v2.3).

B. User Pace Matching to Music Tempo

In the first experiment, we evaluated the performance of the pace recognizer. In the experiment, users are required to catch up with music tempos of predefined playlist. Based on the average SWPM, we generated music clips having dynamic BPM ranging from 110 to 170 and played to them. We defined three different types of BPM modes: fixed, gradual and stepwise. We generate music clips according to the BPM modes and then asked subjects to walk or run to the music clips. Fig. 4 (a) shows the result of pace recognition under fixed BPM. Fig. 4 (b) shows the result of user pace recognition under gradual BPM. Finally, Fig. 4 (c) shows the result of user pace recognition under stepwise BPM. To evaluate their accuracy, we calculate their average difference using the following equation.

$$AvgDifference = \frac{\sum_{i \in SWPM\ samples} \left| SWPM_i - (bpm/2) \right|}{\#\,of\ SWPM\ samples} \qquad (4)$$

In addition, we measured their standard devation and minimum -maximum differences. The results are shown in Table I. SWPM+MA is the result of moving average (MA) window after SWPM measurement.

TABLE I. SWPM DIFFERENCES MEASURED

	Average Difference	Standard Deviation	Min ~ Max Difference
SWPM	2.05%	1.654	-49.5 ~ 15.0
SWPM+MA	1.88%	0.905	-20.7 ~ 12.1

C. Pace Recognizing Conditions

Since our proposed scheme aims to assist exercise or workout, its robustness to the external conditions is very important. To this end, we evaluated the accuracy of pace recognition under three different conditions: smartphone is held in exerciser's hand, armband and pocket. Subjects are asked to keep pace 85 SWPM while jogging. Fig. 5 shows the SWPM tracking results of three different conditions. Table 2 shows their comparison in terms of average SWPM difference.

TABLE II. COMPARISON OF RECOGNITION CONDITIONS

	Average SWPM difference (85 SWPM)
Hand	1.74%
Armband	1.35%
Pocket	4.51%

As shown in Fig. 5 and Table II, the proposed scheme can recognize user pace quietly correctly in all conditions. Even with the device in the pocket, our scheme can recognize user pace quite accurately. This is interesting because this is not related to user's swinging action. According to the experiment result, we got the best pace recognition accuracy using the armband.

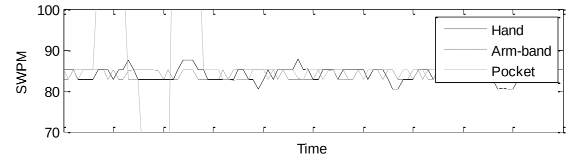

Figure 5. SWPM tracking result in different recognition conditions

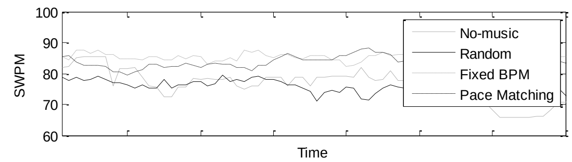

Figure 6. SWPM tracking result of a subject

D. Song Playing Method

In this experiment, we measured the effect of music playing on the exercise or workout. Subjects are asked to keep their own pace as much as possible while jogging 2km course. For the comparison, we considered 4 different jogging conditions: no music, randomly selected music, fixed bpm music and user pace matching music. The measurement is carried out every other day for each subject. Fig. 6 shows SWPM tracking result of a subject for 4 different conditions. Table 3 shows their average SWPMs and the standard deviation (STD).

TABLE III. AVERAGES AND AVERAGE STANDARD DEVIATIONS OF SWPM

	SWPM	SWPM STD
No music	78.8	4.33
Randomly selected music	75.8	3.81
Fixed bpm(170) music	85.4	1.17
Pace matching music	85.1	1.91

From the result, we can observe that controlled music plays such as fixed bpm and pace matching give better exercise effect in term of SWPM. Also, they show less variation than randomly selected music or no music. This means that music can be used to maintain or keep up the effectiveness of exercise or workout.

V. CONCLUSION

In this paper, we proposed Runner's Jukebox, a smartphone music player to improve the user's experience and performance of typical exercises such as walking and running. Based on user pace detection and music feature analysis, our proposed music player retrieves songs and adjusts their playback speed according to the detected user pace. We presented a practical implementation on an Android platform. In addition, for seamless music playback, we proposed music mixing methods. To evaluate the effectiveness of our system, we performed several experiments and showed that our system can work on Android platform with accurate recognition and positive effect on user workout.

ACKNOWLEDGMENT

This work was supported by Basic Science Research Program through the Ministry of Education of the Republic of Korea and the National Research Foundation (NRF) of Korea (NRF-2013R1A1A2012627)

REFERENCES

[1] G. Tenenbaum, R. Lidor, N. Lavyan, K. Morrow, S. Tonnel, A. Gershgoren, J. Meis, and M. Johnson, “The effect of music type on running perseverance and coping with effort sensations,” Psychology of Sport and Exercise, vol. 5, no. 2, pp. 89–109, Apr. 2004.

[2] C. I. Karageorghis and P. C. Terry, “The psychophysical effects of music in sport and exercise: a review,” Journal of Sport Behavior, vol. 20, no. 1, pp. 54–68, 1997.

[3] F. Styns, L. van Noorden, D. Moelants, and M. Leman, “Walking on music,” Human Movement Science, vol. 26, no. 5, pp. 769–785, Oct. 2007.

[4] G. T. Elliott and B. Tomlinson, “PersonalSoundtrack: context-aware playlists that adapt to user pace,” CHI ’06 Extended Abstracts on Human Factors in Computing, pp.736–741, Apr. 2006.

[5] J. A. Hockman, M. M. Wanderley, and I. Fujinaga, “Real-Time Phase Vocoder Manipulation by Runner’s Pace,” Proceedings of the International Conference on New Interfaces for Musical Expression, pp. 90–93, June 2009.

[6] G. Wijnalda, S. Pauws, F. Vignoli, and H. Stuckenschmidt, “A Personalized Music System for Motivation in Sport Performance,” IEEE Pervasive Computing, pp. 26–32, 2005.

[7] B. Moens, L. van Noorden, and M. Leman, “D-Jogger: Syncing Music with Walking,” Proceedings of SMC Conference 2010, Barcelona, pp. 451–456, 2010.

[8] N. Oliver and F. Flores-Mangas, “MPTrain: A Mobile, Music and Physiology-based Personal Trainer,” Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services, pp. 21–28, 2006.

[9] S. Dixon, “Evaluation of the Audio Beat Tracking System BeatRoot,” Journal of New Music Research. vol. 36, no. 1, pp. 39–50, 2007.

[10] SoundTouch Sound Processing Library. Available from: http://www.surina.net/soundtouch/. [retrieved: Feb. 2015].