Keep the Beat: Audio Guidance for Runner Training

Luca Balvis, Ludovico Boratto, Fabrizio Mulas, Lucio Spano, Salvatore Carta, Gianni Fenu

Keep the Beat: audio guidance for runner training

Luca Balvis, Ludovico Boratto, Fabrizio Mulas,
Lucio Davide Spano, Salvatore Carta, and Gianni Fenu[1]

University of Cagliari,
Department of Mathematics and Computer Science,
Via Ospedale 72, 09124, Cagliari, Italy
{luca.balvis, ludovico.boratto, fabrizio.mulas,
davide.spano, salvatore, fenu}@unica.it

Abstract. Understanding how to map the feedback by fitness apps into concrete actions during the exercise performance is crucial for their effectiveness, for both inexperienced and advanced users. In this paper we focus on audio feedback for running, describing a beat-rhythm representation of the target cadence for helping the user in keeping it. We designed the feedback system in order to balance two conflicting objectives: its effectiveness in helping the user in reaching the training goal and its intrusiveness with respect to concurrent activities (e.g., listening to the music). We detail how we track the user's cadence through standard smartphone sensors, how and when we generate the audio messages. Finally, we discuss the results of a user-study, showing effectiveness with respect to the adherence to the exercise goal and the overall usability.

Keywords: Audio guidance; Beats; Running; Training; Evaluation; Fitness;

1 Introduction

Smartphones have embedded sensors of increasing quality through the years, which quickly substituted entry-level devices for different tasks, such as e.g. taking pictures, videos, recording audio, browsing the internet. In particular, inertial sensors (accelerometers, gyroscopes, compass) and GPS receivers constituted the basis for the entry level activity tracker, which is now represented by different smartphone apps. The activity tracker applications are now spread also among the entry-level users, since they are very cheap, if compared with professional devices and applications.

Opening the market to such kind of users is a big opportunity for fitness application providers, but it is also a big challenge: the design of dedicated devices was targeted to professional users, which need many precise measures they are able to interpret directly. Many app users are not experts instead, so they would need a more intuitive representation of the tracked data.

If we focus on the information needed *during* the exercise, this is even more difficult: the user not only has to understand what the tracked data means, but she has also to take decisions based on such data for continuing the session. In this paper we consider the running activity as an example of the latter case: while running, the user should keep a target cadence, and smartphone applications help her tracking and reporting its current value. During the training, the application cannot rely on the visual channel, so audio messages are usually preferred. However, many runners are used to listen to the music or other entertainment content that may require the audio channel (e.g., audiobooks). Therefore, it is important for the feedback system to provide the relevant information without prevent the user to carry-out such entertainment activities and taking the complete control of the audio channel [11,13].

Considering the information the system provides to the user, reporting the current cadence is useful, but what if its value is not in line with the target? Entry-level users may have difficulties in mapping the application reported difference between the target and the current cadence into a concrete running action. Professional users have less difficulties in interpreting the values, however they still need to repeatedly check the measures and to take decisions accordingly.

In this paper we describe a generic audio feedback engine for runners, whose architecture allows providing adaptable audio feedback, exploiting the audio channel only when needed. It is possible for designers to control the number and the type of the generated feedback messages, while users may override the notification frequency in order to fine-tune it to their needs. The feedback system relies only on built-in smartphone hardware for tracking the activity. We describe the software component that senses the user's cadence, together with the rule system that decides how and when the application generates the audio messages. We detail how we configured it for providing a beat-rhythm representation of the correct cadence, which helps the user in keeping the target for the current training. In order to keep it unintrusive, the beats are played only if the user is running out of range.

Finally, we provide the results of a user-study having a twofold objective. The first one was to compare the effectiveness of the beat feedback in helping the user in keeping the target pace, comparing it to providing natural language messages. The second goal was to collect a qualitative data on different usability dimensions for the proposed feedback, showing both a very good perceived precision and completeness for provided information.

2 Related work

Mobiles have been used as activity trackers even before they were equipped with inertial sensors [1]. Since then, the tracking techniques have evolved and now it is possible to track the user's steps with a good level of precision using inertial sensors such as accelerometers [12] or gyroscopes [4], which are included in majority of smartphone models nowadays. For tracking the user's cadence, we built our prototype on top of such approaches, as many other commercial apps.

Even though several academic studies have been performed to support users in their running activities, finding applications that work in the real world and have an audio support is rarer. The three platforms that involve most users, i.e., *Endomondo*[1], *Runtastic*[2], and *Nike+*[3], all offer a form of audio support in which it is possible for the user to let the application read statistics when she covered a given distance or a specific amount of time has passed. These statistics involve common information, such as the average pace, speed, calories burned, heartbeat, etc. A form of support that provides the user with indications about the speed is provided by *u4fit* (previously known as Everywhere Run) [6]. Differently from the other applications, it checks the current pace of the user and compares it with the objective she had, and plays an audio support that tells the user to speed up or slow down.

The topic of providing effective audio feedback to users has been investigated under different perspectives in literature, especially considering mobile devices, which are often used while carrying-out different tasks. For this reason, different audio techniques have been developed for helping users in making selections without looking to the mobile screen [15]. Considering the particular application of audio feedback for providing guidance during physical activities, it is possible to find examples for different sports in literature. For instance, Nylander et al. [8] created an audio feedback for guiding the user in reaching the highest point of acceleration just before hitting the ball. Stienstra et al. [10] used a sonification technique for guiding professional skaters with a feedback on their performance in real time.

Considering running applications, we can find examples of auditory feedback in different work in literature. MPTrain [9] includes a set of sensing devices connected to a mobile phone, that are able to sense both the user's pace and physiological parameters, exploiting music for helping the user in achieving her goal. Our work focuses on built-in smartphone hardware and exploits an explicit feedback rather than on an implicit song selection. An evolution of this work, TripleBeat [2], introduced additional features like competition and a visual interface for motivating the user during the exercise.

Song playlists may be used as implicit feedback for helping users in keeping the target cadence. Indeed, music has different effects in the runner's performance, such as the synchronization of the cadence with the song tempo [11] and an overall higher performance and less perceived fatigue [13]. For instance, *RockMyRun*[4] is a commercial running application that selects songs from a playlist according to the user's hear rate, steps or target cadence, in order to increase the motivation and her performance. Even if this has been proven to be an effective support, it has two main drawbacks. Firstly, if the user is not willing to listen to the music during the exercise, she cannot receive the application feedback. In addition, it may happen that the user would like to listen to specific song that maybe do not match with the feedback requirements. In contrast, our feedback

[1] https://www.endomondo.com/
[2] https://www.runtastic.com/
[3] https://secure-nikeplus.nike.com/plus/
[4] http://www.rockmyrun.com/

mechanism exploits the audio channel only when needed and not for the whole duration of the exercise.

PaceGuard [3] exploits a pulse feedback, similar to the one exploited in this paper, played during the whole exercise and adapted to the current user's performance. Differently from PaceGuard, we provide the audio feedback only if the cadence is out of range. In the meantime, other smartphone applications may use the audio channel (e.g., music players, voice calls). Considering smartphone-based applications, the audio feedback has been exploited for providing a remote support for encouraging users during races [14]. RunRight [7] creates a visual representation of the running movement using the acceleration on the vertical and horizontal axis. Similarly to the approach proposed in this paper, the system provides also a metronome for communicating the running cadence to the user, which changed randomly after a predefined amount of time. Our approach combines the information coming from the inertial sensors with the audio feedback for helping the user in reaching the training goal.

3 Audio support

In our work, we considered a scenario where the user should keep a target cadence, which is defined in advance according to the user's training goal (e.g., preparing for a competition) and the intensity not exceeding her capabilities. The target may be selected with the help of a trainer or by the user herself, but the application does not suggest it, since without heart-rate measures it would not be able to determine the correct intensity and this may lead to heart problems.

We designed the audio feedback support considering the trade-off between two aspects: the first one is the need to guide the user and to help her in keeping the correct cadence, while the second one is the intrusiveness of the support. On the one hand, sending messages each time we register a high difference between the current and the expected cadence guarantees the quickest possible user's reaction. On the other hand, if the user has difficulties in keeping the cadence, the feedback would be frustrating and it would take the complete control of the audio channel, preventing the users to perform other entertainment activities while running (e.g., listen to the music, radio or audio books).

Therefore, we designed a support able to adapt its behaviour to the user needs, which is configurable through different parameters, each one having a default value, which may be changed by the user for fine-tuning. The parameter set is the following:

- **Sampling interval** (i_s): the frequency for calculating the current cadence. The default value is 1 minute.
- **Target cadence** (k): the cadence that our user should keep during the current activity (or the current part of it), expressed in terms of steps per minute.
- **Tolerance.** (ϵ): the number of steps per sampling interval that can be tolerated if a user is above or under the desired cadence, in order to consider

her current cadence as correct. This allows us to define a tolerance range, as $k \pm \epsilon$. The ϵ default value is 5.

- **Feedback message duration.** (t_f): the number of seconds the audio message should last. The default value is 15s.
- **Frequency of positive feedback messages** ($\frac{1}{i_p}$): the number of minutes that should pass between an audio message and the next, when the user's cadence is correct. The default value for i_p is 4 minutes.
- **Frequency of negative feedback messages** ($\frac{1}{i_n}$): the number of minutes that should pass between an audio message and the next, when the cadence of the user is out of range.The default value for i_n is 1 minute.

The audio feedback module contains two main components, separating the cadence tracking from the set of rules for generating the feedback messages:

1. **Cadence analyser.** This component analyses the current cadence of the user, in order to decide if it is correct, high or low.
2. **Feedback manager.** Given the decision of the cadence analyser and the current settings, this component decides whether a feedback message should be sent to the user and, if this is the case, it is responsible for creating it.

In the next sections, we describe in detail the two components.

3.1 Cadence Analyser

Given the current cadence of the user, i.e., the number of steps done in the last minute, this component determines if an audio message to support the user should be played, according to the constraints given as input to the application. It tracks the user's cadence exploiting the Step Counter Sensor[5] provided with the default Android SDK, which internally exploits the accelerometers.

The cadence analyser component is a simple state machine, depicted in figure 1. It classifies the user's cadence into three classes, corresponding to its internal states:

1. *Correct*, when the user's cadence v is inside the target range $k \pm \epsilon$;
2. *High*, when the user's cadence v is higher than the maximum of the range $k + \epsilon$;
3. *Low*, when the user's cadence v is lower than the minimum of the range $k - \epsilon$;

The state transitions are defined through the value of Δ, which we calculate as follows (v is the current user's cadence)

- if $|k - v| < \epsilon$, then $\Delta = 0$;
- if $v > k + \epsilon$, then $\Delta > 0$;
- if $v > k - \epsilon$, then $\Delta < 0$;

[5] http://developer.android.com/guide/topics/sensors/sensors_motion.html

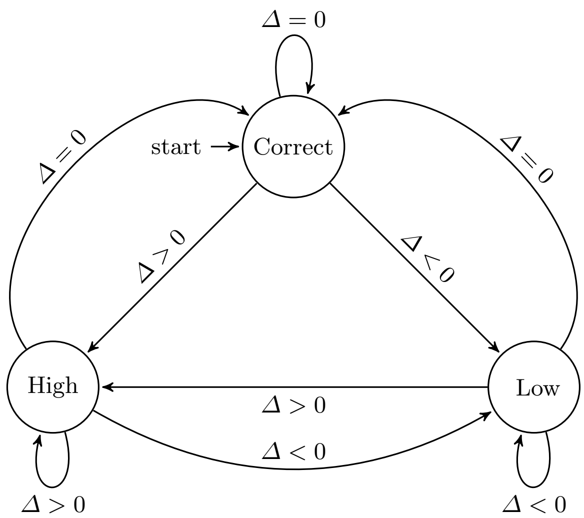

Fig. 1. The Pace Analyser state machine

In each state, if $\Delta = 0$ the machine fires a transition towards *Correct*, if $\Delta > 0$ the next state is *High*, otherwise the next state is *Low*.

Besides its internal state, the component exposes a variable Δt, that tracks how long the machine is in the current state. This variable is increased of i_s each time the machine fires a transition towards the same state and reset to zero each time the transition is towards a different state. The component raises an event each time a transition is fired, passing as arguments the current state s and Δt.

3.2 Feedback Manager

The feedback manager component registers to the events raised by the cadence analyser, and it contains a set of rules for generating the audio feedback. In the current configuration, we included three rules, which we summarise in Table 1.

The first rule (line 1) triggers while the user is running for a long time (longer than i_n) at a high cadence (s is *High*). It resets Δt to zero, renders a speech message for informing the user that her current cadence is too high, i.e. "Your cadence is too high. Try to follow this rhythm:" (line 3) and generates the beat feedback (line 4). The rules handling the other two cases, respectively low cadence (line 5-8) and current cadence (line 9-12) are similar: they render a different message through the text-to-speech, and we consider a different time in the correct case (i_p).

As already discussed while introducing the paper, we represent the cadence information as a rhythm, in order to allow the user to synchronize her steps with the beats. The PlayBeats procedure generates such feedback on the fly, creating and playing an audio stream containing a number of beat samples equally

upon event $(s = \text{High}, \Delta t > i_n)$ **do**
  $\Delta t := 0$
  TTS(DecreaseSpeedMsg)
  PLAYBEATS(k, t_f)
upon event $(s = \text{Low}, \Delta t > i_n)$ **do**
  $\Delta t := 0$
  TTS(IncreaseSpeedMsg)
  PLAYBEATS(k, t_f)
upon event $(s = \text{Correct}, \Delta t > i_p)$ **do**
  $\Delta t := 0$
  TTS(OkMsg)
  PLAYBEATS(k, t_f)

Table 1. Rules for triggering audio feedback in the Feedback manager module.

distributed in its duration. Such number n is obtained according to two parameters: the target cadence k and feedback message duration t_f, as defined in equation 1:

$$n = \frac{1}{60} \cdot k \cdot t_f \tag{1}$$

The peak of each beat is positioned exactly in each n-th fraction of t_f. The duration between two beats is filled with silence. The resulting message contains a simple click with the rhythm to be followed by the user. After t_f seconds (15 by default), the audio channel is released by the feedback manager and it can be used again by other applications (e.g., the music player).

3.3 Discussion

We would like to point out here a set of relevant properties of the proposed audio feedback support. The first one is its flexibility: the separation of the two components makes it possible to provide different kinds of messages without changing the cadence tracking algorithm and vice versa. By maintaining the same communication protocol, it would be possible to exploit more sophisticated strategies for deciding if the cadence of our user is correct or not. The second important aspect is the possibility to configure the feedback messages through a set of rules, which may be defined differently by interface designers for increasing the effectiveness of the messages, and obtaining different types of feedback. As an example, we can consider the audio guidance provided by *u4fit* [6]. It exploits text-to-speech messages for informing the user in natural language, providing also quantitative information with messages such as e.g. "You're 10s per km slower, speed up!". Such messages may be easily generated modifying the rules in Table 1, using $k-v$ as parameter for both the text to speech messages (increase ad decrease) and removing the calls to PLAYBEATS.

It would be possible to support a completely different feedback mechanism, such as for instance selecting a song from a playlist having a tempo that helps the user in increasing or decreasing her cadence, similarly to *RockMyRun*[6]. According to the difference between the current and the target cadence $(k - v)$, the rules in Table 1 would enqueue a song with a slower, faster or similar tempo for respectively decreasing, increasing and keeping constant the user's cadence.

The rules are also quite straightforward to understand even for users, if represented in natural language. This is very important, in order to enable the user's control over the feedback mechanism: if users have means for inspecting the rules and modify them (through guided procedures), it is more likely that they will be able to get the feedback they need. However, we are also aware that the large majority of the users are not willing to configure such aspects, therefore a good default mechanism is still crucial.

Finally, the parameters used for tuning the feedback and the run-time generation of the messages allows to exploit them in complex workouts, composed by different phases where the user should keep different cadences, or containing an adaptation of the target according to the current user's performance.

4 Evaluation

A user test has been carried out in order to evaluate the ability of the auditory support to guide the users in keeping a constant cadence during the workout, while maintaining the feedback unintrusive. Therefore, as baseline approach for providing feedback, we selected the approach among the currently available applications that we considered the most informative while keeping an unintrusive approach, which is the one used by *u4fit*. It exploits text-to-speech messages for informing the user in natural language, providing also quantitative information with messages such as e.g. "You're 10s per km slower, speed up!".

It is worth pointing out that the experiment setting is different from the studies by Terry et al. [11] and by Waterhouse et al. [13], since in these studies the music stimulus was available during the whole exercise, causing the reported synchronization effect. Instead, we provide feedback messages (either in natural language or through the beats), only when the user is out of range or, if the user is in range, we provide confirmation feedback at regular time intervals. We study the effectiveness of two different feedback types in this context.

Besides the comparative study, we included in the evaluation a set of qualitative aspects for understanding the overall user experience with the proposed support, in order to assess whether unintrusive feedback techniques are acceptable for users.

4.1 Test Design

In order to perform the comparison between the two feedback types, we created two different prototypes, one providing feedback through voice (V) and one

[6] `http://www.rockmyrun.com/`

through the beats (B). In the V condition we configured the rules for obtaining the feedback used in *u4fit*, while in the B condition we used the rules described in this paper, with the default values for the parameters.

We recruited the participants among the students of the University of Cagliari, through both board notices and social networks, asking for people already following a training programme in running (at any level). Before starting the test, each user filled a demographic questionnaire and read a description of the experiment aim and organization. We explained them that the goal of the test was to evaluate the effectiveness of two types of feedback for helping them in keeping a constant cadence during the workout.

After that, each user started with a calibration phase, where we asked them to run at a sustainable cadence. We measured the user's cadence during a workout without providing any feedback and, in order to set a target speed that would be sustainable for him/her, we took the median value of the session. The cadence was measured using the same step-counter included in the two prototypes.

For the comparison phase we used a within-subject design: each user tried both prototypes during two different workouts. In order to avoid the carry-over effect, half of the users started from condition V and half of them started from condition B. In both conditions we set the same target cadence, which we obtained during the calibration.

During the workout, the application logged the current cadence, expressed in number of steps per minute. With this data we were able to calculate the difference between the target and the current cadence for each minute of the workout, whose duration was fixed to 20 minutes.

At the end of each workout condition, we requested them to rate the following aspects of the audio feedback, in a 1 to 7 Likert scale (1 strongly disagree, 7 strongly agree). We also report the question asked:

- **Usefulness**: Please rate how useful the audio feedback was (1 useless, 7 very useful);
- **Timeliness**: Please rate how timely the audio feedback was (1 useless, 7 very useful);
- **Completeness**: I had all the information I needed.
- **Effort**: I did not have to reason on the feedback information in order to understand what it meant.
- **Satisfaction**: I was satisfied by the provided information.
- **Precision**: The provided information was precise.

In addition, we included an open-ended question for collecting suggestions and/or observations on the audio feedback.

4.2 Test Results

Twelve persons participated to the test, nine males and three females, their age ranged from 23 to 31 years old ($\bar{x} = 27.33$, $s = 2.46$). Most users have an average experience with running: three of them have a training once a month, one once every two weeks, two once a week, five twice a week and one once every two days. The experience with apps for running was quite low: half of them never used them, two use it once every ten trainings, while four use apps during every session. Instead, the experience with smartphone is high for all users, all of them use apps more than once a day.

In order to evaluate the effectiveness of the two designs, we measured how good it would be to fit the recorded cadence of the user during each session with the constant value obtained from the calibration phase. In order to measure the fitting quality we used the root-mean-square error (RMSE), defined in equation 2 as the standard deviation of the difference between the target cadence c and the actual user's cadence p_t at minute t (n is the duration in minutes of the workout).

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(c - p_t)^2}{n}} \quad (2)$$

We registered a higher value of the RMSE in the vocal condition ($\bar{x}_V = 4.84$, $s_V = 3.79$), while in the beat condition we registered both a lower central and spread values ($\bar{x}_B = 2.33$, $s_B = 0.50$). The paired t-test highlighted a significant difference between the two means ($t(11) = 2.23$, $p < 0.05$), and the 95% confidence interval of such difference is 2.50 ± 2.48.

This shows that the beat feedback is more effective for users and that they are able to better control their cadence while running also in the case feedback is provided only when they are out of range. The users were able to synchronize their cadence with the beats even if the stimulus was not continuous during the whole exercise.

The results of the qualitative assessment are shown in figure 2. The boxplot shows that the ratings expressed for the beats version are consistently higher than those for the voice version. In particular, we found a significant difference for the usefulness ($t(11) = 2.24$, $p < 0.05$, $CI = [0.02; 2.15]$) and a practical significance for the precision ($t(11) = 1.98$. $p < 0.07$, $CI = [-0.13; 2.48]$). It is also important to notice that the timeliness of the feedback received good ratings and there is no perceived difference between the two conditions (as expected, since the Cadence Analyser had the same settings in both cases).

In the open ended questions, the users asked for more feedback when their cadence was in line with the target in both versions. Increasing the number of audio messages is obviously possible, but we should find a good balance between this need and the usage of the audio channel for e.g., listening to the music. Providing too many messages may be annoying in that case. One of the users suggested to increase the number of messages through the time, especially at the end of the training, which would be easily supported with our engine.

Considering the vocal message condition, the users highlighted in the comments some difficulties in understanding how much they should increase or decrease their speed. Once the message communicated that the cadence was not in line with the target, it was not easy for them to articulate their intention into actions. Instead, in the beats condition the user recognized that it was easy to understand how much they should change their cadence. However, they suggested to complete such information with overall information on the training (duration, kilometers covered, average cadence, etc.).

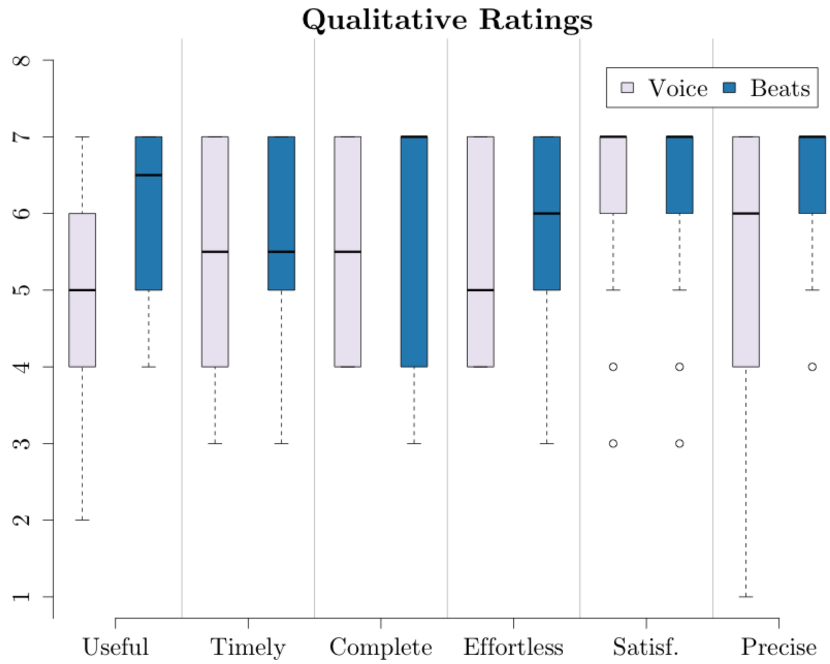

Fig. 2. Qualitative evaluation summary

5 Conclusion and future work

In this paper we discussed an audio feedback engine for providing cadence information to runners while training. We discussed how we tracked the current cadence and the rules for sending audio messages, trying to balance the trade-off between the timeliness and the effectiveness of the information and the disturbance for the user. We reported the results of a user-study comparing this technique with the one currently used in different smartphone apps in the market. Finally we reported a qualitative evaluation of its overall usability.

In future work, we aim to include such feedback system into a more complex training support, where the user may set different cadences for different parts of the workout. In addition, we would like to provide the feedback system to a larger number of users, and extracting different standard default configuration profiles. In addition, we would also enhance the feedback system in order to exploit data coming from more advanced sensors that runners may own (e.g., heart-rate sensors, GPS, etc.).

References

1. Consolvo, S., McDonald, D.W., Toscos, T., Chen, M.Y., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., Libby, R., Smith, I., Landay, J.A.: Activity

sensing in the wild: A field trial of ubifit garden. In: Proceedings of CHI '08. pp. 1797–1806. ACM (2008), http://doi.acm.org/10.1145/1357054.1357335

2. De Oliveira, R., Oliver, N.: Triplebeat: Enhancing exercise performance with persuasion. In: Proceedings of MobileHCI '08. pp. 255–264. ACM (2008), http://doi.acm.org/10.1145/1409240.1409268
3. Fortmann, J., Pielot, M., Mittelsdorf, M., Büscher, M., Trienen, S., Boll, S.: Paceguard: Improving running cadence by real-time auditory feedback. In: Proceedings of MobileHCI '12. pp. 5–10. ACM (2012), http://doi.acm.org/10.1145/2371664.2371668
4. Jayalath, S., Abhayasinghe, N.: A gyroscopic data based pedometer algorithm. In: Proceedings of ICCSE 2013. pp. 551–555 (April 2013)
5. Moens, B., Muller, C., van Noorden, L., Franěk, M., Celie, B., Boone, J., Bourgois, J., Leman, M.: Encouraging spontaneous synchronisation with d-jogger, an adaptive music player that aligns movement and music. PloS one 9(12), e114234 (2014)
6. Mulas, F., Carta, S., Pilloni, P., Manca, M.: Everywhere run: a virtual personal trainer for supporting people in their running activity. In: Proceedings of ACE '11. pp. 70:1–70:2. ACM (2011)
7. Nylander, S., Jacobsson, M., Tholander, J.: Runright: Real-time visual and audio feedback on running. In: CHI '14 Extended Abstracts. pp. 583–586. CHI EA '14, ACM (2014), http://doi.acm.org/10.1145/2559206.2574806
8. Nylander, S., Kent, A., Tholander, J.: Swing sound: Experiencing the golf swing through sound. In: CHI '14 Extended Abstracts. pp. 443–446. ACM (2014), http://doi.acm.org/10.1145/2559206.2574789
9. Oliver, N., Flores-Mangas, F.: Mptrain: A mobile, music and physiology-based personal trainer. In: Proceedings of MobileHCI '06. pp. 21–28. ACM (2006), http://doi.acm.org/10.1145/1152215.1152221
10. Stienstra, J., Overbeeke, K., Wensveen, S.: Embodying complexity through movement sonification: Case study on empowering the speed-skater. In: Proceedings of CHItaly 2011. pp. 39–44. ACM (2011), http://doi.acm.org/10.1145/2037296.2037310
11. Terry, P.C., Karageorghis, C.I., Saha, A.M., DAuria, S.: Effects of synchronous music on treadmill running among elite triathletes. Journal of Science and Medicine in Sport 15(1), 52–57 (2012)
12. Tomlein, M., Bielik, P., Krtky, P., tefan Mitrk, Barla, M., Bielikov, M.: Advanced pedometer for smartphone-based activity tracking. In: Proceedings of BIOSTEC 2012. pp. 401–404 (2012)
13. Waterhouse, J., Hudson, P., Edwards, B.: Effects of music tempo upon submaximal cycling performance. Scandinavian journal of medicine & science in sports 20(4), 662–669 (2010)
14. Woźniak, P., Knaving, K., Björk, S., Fjeld, M.: Rufus: Remote supporter feedback for long-distance runners. In: Proceedings of MobileHCI '15. pp. 115–124. ACM (2015), http://doi.acm.org/10.1145/2785830.2785893
15. Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., Baudisch, P.: Earpod: Eyes-free menu selection using touch input and reactive audio feedback. In: Proceedings of CHI'07. pp. 1395–1404. ACM (2007), http://doi.acm.org/10.1145/1240624.1240836