A Tempo-Sensitive Music Search Engine With Multimodal Inputs

Yu Yi
School of Computing
National University of Singapore
117590 Singapore
YiYu@Ymail.com

Yinsheng Zhou
School of Computing
National University of Singapore
117590 Singapore
yzhou86@comp.nus.edu.sg

Ye Wang
School of Computing
National University of Singapore
117590 Singapore
wangye@comp.nus.edu.sg

ABSTRACT
This paper presents TMSE: a novel Tempo-sensitive Music Search Engine with multimodal inputs for wellness and therapeutic applications. TMSE integrates six different interaction modes, Query-by-Number, Query-by-Sliding, Query-by-Example, Query-by-Tapping, Query-by-Clapping, and Query-by-Walking, into one single interface for narrowing the intention gap when a user searches for music by tempo. Our preliminary evaluation results indicate that multimodal inputs of TMSE enable users to formulate tempo related queries more easily in comparison with existing music search engines.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Query formulation; H.5.1 [**Multimedia Information Systems**]: Audio input/output; H.5.5 [**Sound and Music Computing**]: Signal analysis, synthesis, and processing.

General Terms
Algorithms, Design, Experimentation, Human Factors

Keywords
Multimodal Query, Tempo, Music Information Retrieval

1. INTRODUCTION
Tempo is a basic descriptor of music. The act of tapping one's foot in time to music is an intuitive and often unconscious human response [4]. Content-based tempo analysis is important for certain applications. For example, music therapists use songs with particular tempi to assist the Parkinson's disease patients with gait training. This method, also known as rhythmic auditory stimulation (RAS) [8], has been shown to be effective in helping these patients achieve better motor performance. Music tempo can also facilitate people's running exercises when they follow the music beats during running. It motivates them to run and makes them to feel less tired [14].

MIRUM'11, November 30, 2011, Scottsdale, Arizona, USA.

Figure 1: TMSE User Interface

In above scenarios, a well-designed music search engine for tempo will help users to achieve the goal: users can search for a list of songs based on the tempo information. However, the search box in a traditional search engine constrains the way to express the music tempo in a query. Although it is easier for users with music background (e.g. trained musicians or music therapists) to input a number as BPM (*beats-per-minutes*) value to accomplish their query, the mere number of BPM makes little sense to ordinary users. Most of ordinary users can hardly express it in the form of a rigid number, even when they have their desired tempo to search. The text input mode hampers users' expression of the music tempo that occurs in their mind, which forms so-called **Intention Gap**. Intention gap is a gap between users' search intents and the queries, because of the incapability of key-word queries to express users' intends [21]. Intention gap often makes users unsatisfied with the search results. This motivates us to develop a tempo-sensitive music search engine.

Some critical questions about tempo-sensitive search engine include:

1. How to estimate the tempo of a song?
2. How to design a user interface with more choices that enable users to express their intention easily?
3. How to design a user interface that allows users to verify the tempo values of songs returned by the search engine?

For the question 1 itself, we adopted 4 existing beat tracking methods to estimate tempo followed by evaluation. We chose

the best tempo estimator in our system implementation. This is a part of an ongoing project [15].

The focus of this paper is to address questions 2 and 3. For users to express music tempo with more choices (question 2), we provide users multimodal inputs to express their intention in tempo. The multimodal inputs include (see Figure 1):

- **Query-by-Number:** a user inputs a tempo value in BPM; TMSE returns a list of songs close to that tempo value.
- **Query-by-Tapping:** a user taps the mouse; TMSE returns songs with tempo values close to the user's tapping speed.
- **Query-by-Example:** a user uploads a song; TMSE returns songs with similar tempo of the example song.
- **Query-by-Clapping:** a user claps his/her hands; TMSE return songs matching the speed of clapping.
- **Query-by-Sliding:** a user retrieves songs in different tempi by sliding the tempo bar. This is designed for users without music background to have a sense of quick and slow tempo.
- **Query-by-Walking:** when a user carries an iPhone, TMSE detects the user's walking speed and returns songs close to his/her walking speed.

Choosing Query-by-Number is because it is the most traditional query method. Choosing Query-by-Sliding is because it can add the sense of fast/slow than Query-by-Number, and we would like to know whether it is effective. Choosing Query-by-Tapping and Query-by-Clapping is because they can both act as natural response of human beings towards music tempo, and Query-by-Clapping is even more natural and interesting than Query-by-Tapping. Choosing Query-by-Example is because it is used widely and effectively in other MIR applications. Choosing Query-by-Walking is because the its potential clinical use in the future, and also it is a challenging research problem.

In order to verify search results from TMSE (question 3), the user can listen to the result while clicking the mouse. The estimated music tempo and the clicking tempo will be displayed in the interface side by side thus allowing an easy comparison. To further enhance user experience, we provide users a button near the song name to modify tempo without changing the pitch.

The research contributions of this paper are as follows:

- We have developed a multimodal Tempo-sensitive Music Search Engine (TMSE), to fulfill users' tempo search requirement, which is typically lacking in existing music search engines.
- We have developed an eyes-free application on iPhone, called iTap, to annotate music tempo in order to collect ground truth for our tempo search engine.
- We have conducted a preliminary usability study of the proposed prototype. We have shown that music tempo searchability is an important aspect missing in most existing music search engines.

The remainder of the paper is organized as follows. Related work is reviewed in Section 2. System implementation is described in details in Section 3, followed by system evaluations in Section 4**.** Conclusion and future work are given in Section 5.

2. RELATED WORK

2.1 Query Inputs in Music Information Retrieval

According to [13], most existing music information retrieval systems can perform either very general or very specific retrieval tasks. Recently, researchers in music information retrieval have been trying to use multiple novel input modalities to fill users' intention gap. For example, Shazam [11] is a successful Query-by-Example application; Users can record a piece of song as query and the system then retrieves the identical songs from the server. Midomi [12] is an example of Query-by-Humming, where a user can sing into the microphone and the system can search for a song that is similar to the user's humming. Query-by-Tapping [9] system is a content-based music retrieval system that allows a user to tap the mouse or clap into microphone to search based on the rhythmic pattern.

All these are popular query inputs for improving user experience in Music Information Retrieval (MIR). However, none of these is capable of searching for music according to tempo. In our search engine, Query-by-Number, Query-by-Sliding, Query-by-Example, Query-by-Tapping, Query-by-Clapping and Query-by-Walking are integrated into one user interface to facilitate users expression of tempo-sensitive query.

2.2 Beat Tracking Algorithms

Many beat tracking algorithms exist. Due to availability of source codes, we have evaluated the following 4 algorithms: Dixon's *BeatRoot* [2, 3, 4], Davies and Plumbley's [1], Ellis' [6] and Klapuri's [7]. An initial version of our tempo-sensitive music search engine was published in [15], which enables music therapists to search for music using Query-by-Number only. In this paper, we focus on the multimodal inputs of the search engine, and its effectiveness in reducing intention gap.

2.3 Eyes-Free Tempo Annotation Tool

There are many projects on eyes-free applications. Some of them are mobile-based auditory feedback UIs, such as auditory icons from Graver and Smith [17], earcons from Brester et al. [18], and earPod from Zhao et al. [19]. However, none of them are focused on developing eyes-free tools for music annotation.

3. SYSTEM IMPLEMENTATION

3.1 iTap: Ground Truth Collection

For developing a tempo-sensitive music search engine, it is critical to have a ground truth dataset, which has to be manually annotated. It is possible to do the annotation using a PC with keyboard or mouse [20]. However, a PC-based annotation tool forces the subject to sit in front of a computer to accomplish the annotation task with full visual attention, which makes the annotation task boring and stressful.

On the other hand, tapping along with music is an unconscious human response [4]. Therefore full visual attention is not necessary throughout the whole tempo annotation process. This motivated us to develop an eyes-free annotation tool, iTap, on mobile devices, which provides the following benefits:

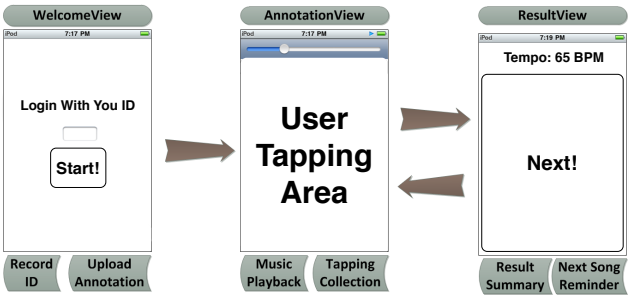

Figure 2: iTap User Flow

1. An eyes-free tempo annotation tool allows the subject to perform the annotation task anywhere and anytime. For example, the subject can do the annotation while walking or commuting in a bus or subway with little visual attention.
2. We intend to extend the iTap to become a GWAP (Game-With-A-Purpose) in order to motivate more subjects to annotate our music database.

Figure 2 depicts the user interface and workflow of iTap. Once users submit their IDs, the system generates a music playlist from which they can listen to a 30-second music excerpt randomly chosen from each song. Users start tapping on the screen as long as they can follow the beats of the music while listening. After tapping one piece of music, iTap reminds users to go on to the next one. The tapping area is large enough so that users can tap on it without looking at it. If users want to skip a song, they just slide to the right on the tapping area; if users want to re-play a song, they just slide to the left. Throughout the annotation process, little visual attention is needed. Users only need to wear earphones and listen to the music to tap, even when they are walking or commuting on a bus or subway.

iTap uses the classic client-server architecture, which allows users to annotate anywhere and anytime. When the Internet access is available, they can upload the annotation results onto our server. The client side is written as an iPhone App to collect users' tapping; the server side is written in PHP as a web service running on a Linux server. The server analyzes and stores the tapping data into a MySQL database. iTap stores tapping data locally when the user is tapping, and the data will be uploaded to the server through an HTTP request automatically when the Internet is available. The local tapping data in iTap will be deleted once the upload is successful.

It shows that even lack of professional musical training, most people can tap in time with music, although trained musicians can follow the tempo more quickly and tap in time with music more accurately than non-musicians [5]. Two amateur musicians, both play guitars for more than ten years, were hired to finish the annotation tasks. They were asked to use iTap to annotate whole music dataset. Only the annotation results that matched both musicians would be kept.

3.2 TMSE Search Engine

3.2.1 System Overview

As shown in Figure 1, we have designed the user interface such that users can easily choose a query mode among the six and receive results as a playlist containing songs with similar tempi.

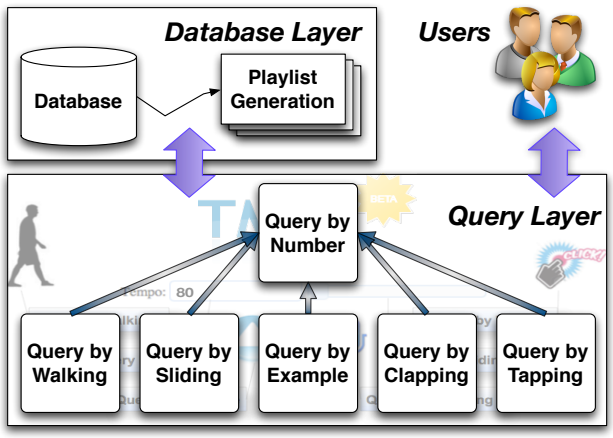

Figure 3: Architecture of TMSE

For the implementation details: TMSE runs under Linux/Unix OS. The UI part of TMSE is developed in HTML5, CSS and AJAX, so that users can enjoy the same user experience under all mainstream web browsers: IE 9, Chrome, and Firefox 5. Two servers are used for TMSE: a web server and a media server. The web server is web.py[1], an open- source web framework based on Python programming language.

The media server is Red5[2], an open-source project for video and audio streaming based on Flash. The reason that a media server is needed is that recording sound from a webpage directly is essential for Query-by-Clapping. It is a hard problem in web development to record audio and video, which has not yet been supported by the HTML5 standard.

Figure 3 illustrates the architecture of TMSE. The system is divided into two layers: the query layer and the database layer. The query layer is used to process different user queries, and to present the results. In the query layer, all the other five query inputs are converted into Query-by-Number, and only BPM values are passed from the query layer to the database layer, regardless the query mode.

The database layer is used to contain the music information, together with their tempo values, which are obtained from the offline processing, using the best-performance beat tracking algorithm in our algorithm evaluation.

3.2.2 Query-by-Number

Query-by-Number is designed to allow users to query songs with an exact tempo value. For Query-by-Number, the tempo value is passed to the database layer and all songs with similar tempi are retrieved into a *VIEW* (a temporal table in database). A playlist is generated from this VIEW, and songs with less tempi difference are ranked higher in the returned result.

3.2.3 Query-by-Sliding

For normal users, the exact tempo value could mean nothing to them, instead *fast/slow* tempi were more important to them. A slide bar is a good indicator of fast/slow, and this is why Query-by-Sliding was implemented as an individual query input in TMSE.

[1] http://webpy.org/

[2] http://www.red5.org/

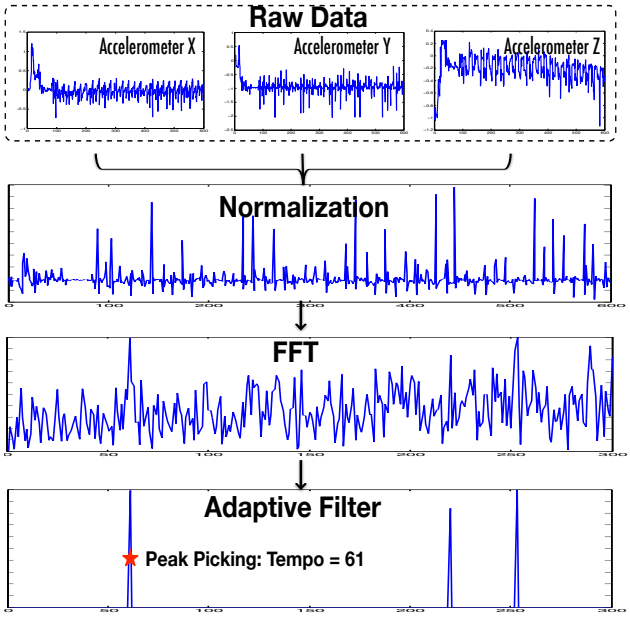

Figure 4: Tempo Estimation Based on Accelerometer Data

For implementation, Query-by-Sliding read the tempo value on the slide bar, and the tempo value is passed to Query-by-Number. When users are sliding, they can see the playlist changes accordingly with different tempi values. Adding auditory feedback to Query-by-Sliding could be a useful feature that remains as a future work.

3.2.4 Query-by-Tapping

Query-by-Tapping allows users retrieve songs according to the speed of users' mouse clicking. For Query-by-Tapping, we used JavaScript on the webpage to capture the movement of users' clicking. The mean value of clicking intervals is used to calculate a tempo value. If a user stops clicking the mouse for more than 1.5 seconds, which corresponds to the slowest tempo of 40 BPM in our database, TMSE will consider it as a boundary between two consecutive queries.

3.2.5 Query-by-Example

Query-by-Example returns songs in the similar tempi of users' example songs. Users can upload an audio file with any audio formats using Query-by-Example. Furthermore, video files with audio channels are also supported. After uploading the file, the server invokes a shell script to convert the audio format into WAV using an open-source tool FFmepg[3]. Then the WAV file is processed using Davies's algorithm [1] to get the tempo value for Query-by-Number.

3.2.6 Query-by-Walking

Query-by-Walking is designed for music therapists to search for music suitable for gait training program of a Parkinson's disease patient. A patient can simply place an iPhone on his/her pockets and walk, and then the tempo of his/her walking could be estimated based on the accelerometer data of the iPhone. This tempo value will then be passed to Query-by-Number.

[3] http://www.ffmepg.org

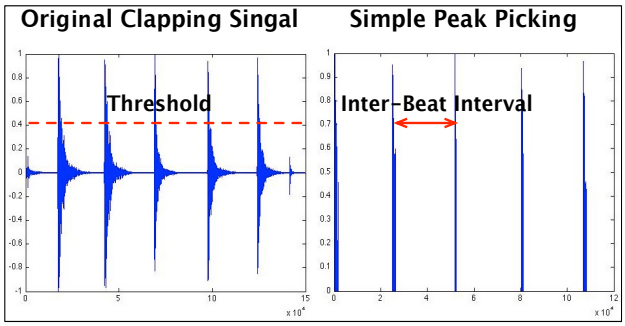

Figure 5: Clapping Signal Processing

Figure 4 shows each step of our algorithm of estimating tempo value from accelerometer data. The accelerometer data from x, y, and z axes are normalized first using quadratic sum:

$$sensor(t) = x(t)^2 + y(t)^2 + z(t)^2$$

The normalized signal is transferred by a FFT routine. We analyze the signal in the frequency domain in order to find out the most dominant frequency, which is corresponding to the frequency in which user walks. The moving average is used as the threshold, and an adaptive filter is applied to the sensor signal. Only the most dominant frequency bins are picked as potential candidates. Then a peak picking method is used to get the maximum of the candidate. The peak picking method make sure that the result frequency is corresponds to a tempo value in the range of 0 - 200BPM, for normally people walk less than 3 steps/second (180BPM). In the example shown in Figure 4, this algorithm outputs a tempo of 61, which is very close to human annotation.

3.2.7 Query-by-Clapping

For Query-by-Clapping, a user can simply clap his/her hands to search for music according to the tempo of his/her clapping. Recording sound directly from the browser is essential for this query input, so we use Red5 as the media server. The Red5 media server stores the streaming data from a microphone to an audio file. After an initial evaluation, we were surprised that all 4 beat tracking algorithms we employed failed in the estimation of tempo clapping for some reasons. For example, algorithms might output 120 BPM for clapping in 60 BPM. As a result, we have designed a simple clapping-picking algorithm on time-domain.

Figure 5 illustrates the thresholding step of our clapping detection algorithm. After normalization, all amplitudes below 40% of the maximum amplitude are set to zeros. We group amplitudes around local maximum as beats, and the clapping tempo is derived from inter-beat intervals. The result is quite reliable in our preliminary study.

4. EVALUATION

Our evaluation consists of two parts: an evaluation on the accuracy of 4 beat tracking algorithms, and a preliminary user study.

Table 1 Accuracy of Tempo Estimation Algorithms

	Davis [1]	Dixon [2,3,4]	Ellis [6]	Klapuri [7]
AC1	0.651	0.492	0.140	0.689
AC2	0.823	0.824	0.646	0.880

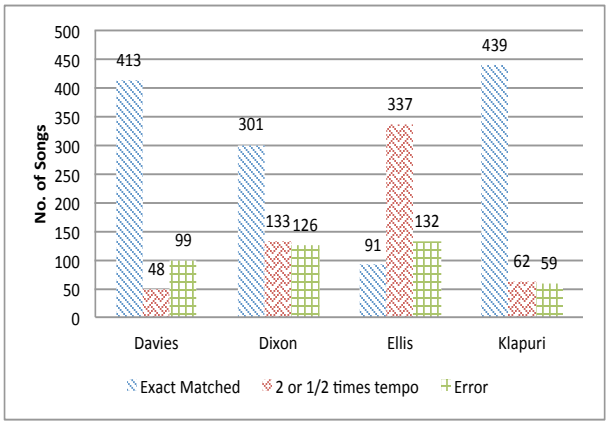

Figure 6: Comparasion between beat tracking algorithms

4.1 Evaluation of Beat Tracking Algorithms

For evaluation of beat tracking algorithms, we collected 787 songs from YouTube, and most of them were old songs because of the potential clinic usage of the system. Two amateur musicians were hired to annotate all songs. Only 560 songs were used as the ground truth after validation.

We compared 4 different beat tracking algorithms on the ground truth. We used each algorithm to get all the inter-beat intervals on each song. We derived tempo value from the mean of the intervals. Results were compared to those in the ground truth.

Table 1 shows the result of the algorithm evaluation. AC1 and AC2 are different accuracy measurements. Let us assume that a is the tempo value of a song obtained from the algorithms, and b is the corresponding tempo value in ground truth. In AC1, only $a = b$ is acceptable. In AC2, three cases are acceptable: i) $a = b$, ii) $a = 2b$, and iii) $2a = b$. In Figure 6, for each of the 4 algorithms, the blue bars illustrates the situation i), the red bars illustrates the situation ii) and iii), and the green bars for errors.

As shown in Table 1, Klapuri's algorithm achieved the best performance among all 4 algorithms. Therefore we chose it for offline processing. For online processing (Query-by-Example), we use Davis' algorithm, because it offers a better tradeoff between accuracy and time efficiency than Klapuri's algorithm.

4.2 Preliminary User Study

4.2.1 Evaluation Setup

We conducted a preliminary user study to validate our system. 15 graduate students volunteered to participate. 9 out of 15 subjects use music search engines (e.g., YouTube) everyday, while the rest use them rarely. We conducted the study in a casual environment, so that users can feel relaxed and give genuine feedback. Evaluation process contains three steps:

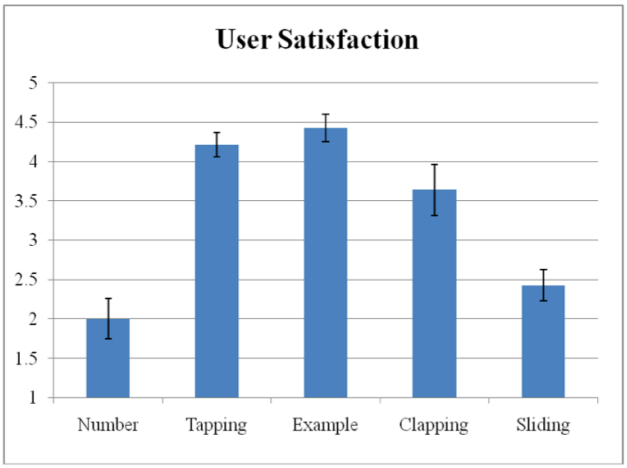

Figure 7: Per-component user satisfaction evaluation

1. **Per-component evaluation[4].** We asked users to listen to one sample song: "Easy Love" by Chinese singer Jay Chou. They were told to feel and remember the tempo of song while listening. Assuming that their intention is to search for songs with similar tempi as the query, they were asked to use each input mode in a random order. After listening to the results returned by search engine, they were asked to rate their satisfaction with each input mode.
2. **Comparison with YouTube.** Since YouTube is one of the most popular online portals with text-based music search capability, we then ask the users to compared YouTube with TMSE in terms of tempo searchability. Therefore, in this study, we want to know whether our tempo-sensitive search can complement some shortcomings of existing search engines like YouTube.
3. **Questionnaire.** All subjects were asked to fill in their basic information such as age, gender, and music search engine experience in a questionnaire. Throughout the whole study, subjects were asked to rate each component or tempo searchability based on the above two tasks.

4.2.2 Result and Analysis

We summarized our evaluation results in Figure 7. In the Figure 7, the highest to lowest ranking for all input modes are Query-by-Example, Query-by-Tapping, Query-by-Clapping, Query-by-Sliding, and Query-by-Number. It is interesting to know that the users' satisfaction with the query mode is proportional to the easiness that they can express the tempo. If they want to express the tempo of a specific song, they prefer Query-by-Example and let the system find the most similar songs. Therefore, it is also not surprising to know that Query-by-Tapping is the second best input mode compared to Query-by-Example, because users can easily tap along with the intended song. Query-by-Clapping is the third best, since clapping users' hands requires more sensory motor efforts than Query-by-Tapping. Query-by-sliding and

[4] Because Query-by-Walking was designed particularly for potential clinic usage of gait training, which is an on-going project. Therefore, we just conducted a preliminary user study for Query-by-Number, Query-by-Tapping, Query-by-Sliding, Query-by-Example and Query-by-Clapping.

Query-by-Number are the lowest two. A possible reason that users were unsatisfied with both two modes is that it was hard for them to express the tempo in the form of rigid number.

We also ask users to compare TSME and YouTube in term of tempo searchability. Users all indicated that TMSE has stronger tempo searchability than YouTube. Since it is hard to indicate tempo in YouTube, users were suggested to input tempo value in the text box during evaluation. We found that some songs returned by YouTube also have the tempo tags in the song titles, which shows the possibility to search by tempo in YouTube in some sense. However, as we showed in the per-component evaluation, Query-by-Number is the hardest way to find the song with a specific tempo. The evaluation results indicate that TMSE could be a complement to existing music search engines.

5. CONCLUSION AND FUTURE WORK

We have presented a tempo-sensitive music search engine with multimodal inputs, Query-by-Number, Query-by-Sliding, Query-by-Tapping, Query-by-Example, Query-by-Clapping and Query-by-Walking, to reduce intention gap when a user formulates a tempo-based query.

We have designed an intuitive and simple-to-use user interface. To validate the proposed prototype, we carried out a preliminary user study consisting of a per-component evaluation and a comparison with YouTube. The results showed that Query-by-Tapping and Query-by-Example were most satisfying and efficient in searching music based on tempo.

We have evaluated 4 beat tracking algorithms, and have selected the best 2 performing algorithms for our system implementation. We have also developed iTap, an eyes-free tempo annotation tool, to annotate our music dataset. We intend to extend iTap as GWAP (Games-With-A-Purpose) in order to collect large amount of annotations.

6. ACKNOWLEDGE

The work was supported by Singaporean MOE grant R-252-000-421-112. The authors would like to thank Dr. Davis, Dr. Dixon, Dr. Ellis and Dr. Klapuri for making their source code available.

7. REFERENCES

[1] Davies, M. E. P. and Plumbley, M. D. Context-Dependent Beat Tracking of Musical Audio, *Audio, Speech and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]* (15:3), 2007, pp. 1009--1020

[2] Dixon, S. Evaluation of the Audio Beat Tracking System BeatRoot, *Journal of New Music Research* (36:1), 2007, pp. 39--50.

[3] Dixon, S. Onset detection revisited, *Proceedings of the 9th International Conference on Digital Audio Effects*, 2006, pp. 133--137.

[4] Dixon, S. Automatic Extraction of Tempo and Beat from Expressive Performances, *Journal of New Music Research* (30), 2001, pp. 39--58.

[5] Drake, C., Penel, A. and Bigand, E. Tapping in time with mechanically and expressively performed music, *Music Perception*, 2000, pp. 1--23.

[6] Ellis, D. P. W. Beat Tracking by Dynamic Programming, *Journal of New Music Research* (36:1), 2007, pp. 51--60.

[7] Klapuri, A. P., Eronen, A. J. and Astola, J. T. Analysis of the meter of acoustic musical signals," *Audio, Speech and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]* (14:1), 2006, pp. 342--355

[8] Thaut, M. H., Mcintosh, G. C., Rice, R. R., Miller, R. A., Rathbun, J. and Brault, J. M. Rhythmic auditory stimulation in gait training for Parkinson's disease patients, *Movement Disorders* (11:2), 1996, pp. 193--200.

[9] Hanna, P., & Robine, M. (2009). Query by tapping system based on alignment algorithm. *ICASSP* 2009, pp. 1881–1884).

[10] Jang, J.S., Lee, H.R., & Yeh, C.-H. (2001). Query by tapping: A new paradigm for Content-Based music retrieval from acoustic input. In H.-Y. Shum, M. Liao, & S.-F. Chang (Eds.) *Advances in Multimedia Information Processing, PCM 2001*, vol. 2195 of *Lecture Notes in Computer Science*, chap. 76, (pp. 590–597).

[11] Wang, A. (2006). The Shazam music recognition service. *Commun. ACM*, *49*(8), 44–48. URL http://dx.doi.org/10.1145/1145287.1145312.

[12] The midomi music search: http://www.midomi.com

[13] R. Typke, F. Wiering, and R. Veltkamp, “A survey of music information retrieval systems,” ISMIR, 2005, pp. 153–160.

[14] Oliver, N. and Flores-Mangas, F. MPTrain: a mobile, music and physiology-based personal trainer. *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, ACM, 2006*, pp. 21—28.

[15] Li, Z., Xiang, Q., Hockman, J., Yang, J., Yi, Y., Fujinaga, I., & Wang, Y. (2010). A music search engine for therapeutic gait training. In *Proceedings of the international conference on Multimedia*, MM ’10, (pp. 627–630). New York, NY, USA: ACM.

[16] “WaoN: A wave-to-notes transcriber,” [Web site] 2006, [2008 Nov 20], Available: www.kichiki.com/WAON

[17] Gaver, W. and Smith, R. Auditory icons in large-scale collaborative environments, *ACM SIGCHI Bulletin* (23:1), 1991, pp. 96.

[18] Brewster, S., Wright, P. and Edwards, A. An evaluation of earcons for use in auditory human-computer interfaces. *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, ACM, 1993, pp. 222—227.

[19] Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R. and Baudisch, P. Earpod: eyes-free menu selection using touch input and reactive audio feedback, *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 2007, pp. 1395—1404.

[20] Brodsky, W. (2005). The effects of metronomic pendular adjustment versus tap-tempo input on the stability and accuracy of tempo perception. *Cognitive Processing*, *6*(2), 117–127.

[21] X. S. Hua, M. Worring, and T. S. Chua, “Internet multimedia search and mining”, *Bentham Science Publishers*, 2010.