# Bridging the User Intention Gap: an Intelligent and Interactive Multidimensional Music Search Engine

Shenggao Zhu[1], Jingli Cai[2], Jiangang Zhang[2], Zhonghua Li[2], Ju-Chiang Wang[3], and Ye Wang[1,2]
[1]NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore
[2]School of Computing, National University of Singapore, Singapore
[3]Department of Electrical and Computer Engineering, University of California, San Diego, USA
shenggaozhu@nus.edu.sg; jingli@comp.nus.edu.sg;
{zhang.agang, lzhlynn, asriver.wang}@gmail.com; wangye@comp.nus.edu.sg

## ABSTRACT

Music is inherently abstract and multidimensional. However, existing music search engines are usually not convenient or too complicated for users to create multidimensional music queries, leading to the intention gap between users' music information needs and the input queries. In this paper, we present a novel content-based music search engine, the Intelligent & Interactive Multidimensional mUsic Search Engine (i$^2$MUSE), which enables users to input music queries with multiple dimensions efficiently and effectively. Six musical dimensions, including tempo, beat strength, genre, mood, instrument, and vocal, are explored in this study. Users can begin a query from any dimension and interact with the system to organize the query. Once the parameters of some dimensions have been set, i$^2$MUSE is able to intelligently highlight a suggested parameter and gray out an un-suggested parameter for every other dimension, helping users express their music intentions and avoid parameter conflicts in the query. In addition, i$^2$MUSE provides a real-time illustration of the percentage of matched tracks in the database. Users can also set the relative weight of each specified dimension. We have conducted a pilot user study with 30 subjects and validated the effectiveness and usability of i$^2$MUSE.

## Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation, Search process; H.5.5 [**Sound and Music Computing**]: Systems

## General Terms

Design, Human Factors, Algorithms, Experimentation

## Keywords

Intention gap; Graphical query interface; Query suggestion and un-suggestion; Query weights; Feedback functions


*WISMM'14,* November 7, 2014, Orlando, Florida, USA.
Copyright 2014 ACM 978-1-4503-3157-9/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661714.2661720.

## 1. INTRODUCTION

With the explosive growth of musical content online, effective music retrieval is itself becoming an art, and the research community continues on the quest for better retrieval techniques or novel search engines [1, 2, 3]. For common users, the first and foremost task in music search is to express their high-level music information needs in a specific form of queries that can be accepted by search engines. However, this is not a trivial task. Because people can perceive music through various dimensions, such as tempo, genre and mood, their music information needs naturally involve multiple dimensions. As an everyday example, a user may want some male-vocal rock songs with strong rhythm to listen to while jogging. From the outset, he may not be familiar enough with musical terminologies to adequately describe what he has in mind to the search engine without a list of options to choose from. Moreover, in case of a traditional text-based music search engine such as YouTube[1], he then will have to describe the multidimensional intention to the search box using a long string of words. Between the user's search intention and what he submits as the query, he is caught in the "intention gap" [4, 5], due to the difficulty of converting the intention into a search-engine-friendly query. The user intention gap remains a major obstacle to meeting the music information needs of users.

Currently several popular music websites such as Last.fm[2], Allmusic[3] and Melodyloops[4] already support music discovery by browsing through a list of tags or categories of a certain dimension. Users can click on what they are interested in to filter the playlist. However, these websites do not support playlist filtering/refining that combines categories from multiple dimensions, and multidimensional queries still have to be input in text.

A few multidimensional music search engines (MMSEs) have been proposed by researchers that support multidimensional queries directly on a graphical interface [1, 6, 7, 8, 9]. For example, MuMa[5] includes dimensions such as chord, genre, mood and date [9]. Various categories of genre and mood are visually listed on MuMa's query interface so that users can click on these categories to organize the queries. Musicovery[6] has a graphical mood panel to search by mood, together with genre and tag queries.

[1]http://www.youtube.com
[2]http://www.last.fm
[3]http://www.allmusic.com
[4]http://www.melodyloops.com/music/
[5]http://muma.labs.exalead.com
[6]http://musicovery.com

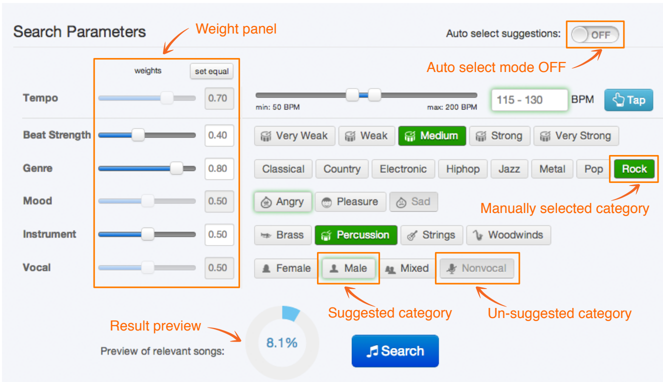



**Figure 1: The query interface of i$^2$MUSE.**

However, existing MMSEs still fail to address the complexity of multidimensional music query and suffer from several drawbacks. First, without intelligent query suggestion, the variety of choices and combinations among dimensions often complicates the search process, making the user experience less straightforward and possibly widening the intention gap; users usually need a lot of time and operations to finalize a query. Second, due to the lack of guidance or real-time feedback on the interface, users may choose potential conflict parameters and end up with very few matched tracks. Third, without a way to specify the relative importance of different dimensions, users have little control over how much weight each of their choices has on the search.

In existing Information Retrieval (IR) systems, techniques such as query suggestion and query expansion have been shown to be effective in refining the user's query and potentially satisfying the user's intention [10, 11, 12]. However, these techniques are limited to retrieval applications based on textual queries. It is not straightforward to apply them to multidimensional music retrieval directly.

### Proposal of i$^2$MUSE

Motivated by the above observations, we propose an Intelligent & Interactive Multidimensional mUsic Search Engine (i$^2$MUSE) to bridge the user intention gap[7]. The novel query interface of i$^2$MUSE is shown in Figure 1.

The main features of i$^2$MUSE are summarized as follows:

• *Multidimensional*: Six dimensions (tempo, beat strength, genre, mood, instrument and vocal) are directly displayed on the interface for easy query input.

• *Content-based*: Unlike traditional metadata-based music search engines, i$^2$MUSE allows users to search for abstract content-based dimensions.

• *Intelligent*: i$^2$MUSE can instantaneously suggest and un-suggest query parameters to the user given a query input. This not only can prevent parameter conflicts in the query but may also automatically match the user intention.

• *Interactive*: On the query interface, i$^2$MUSE automatically highlights the suggested parameters and grays out unsuggested parameters. Real-time result preview is also illustrated on the interface. This interactive process helps users easily and effectively organize the query.

• *Customizable*: By adjusting the weight of each specified dimension through an intuitive slider bar, users have full control of the priority of each dimension in the query.

[7]Online demonstration video: http://youtu.be/wyFXbUTdW-Y

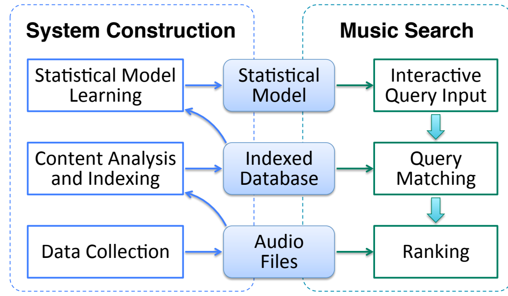



**Figure 2: The framework of i$^2$MUSE.**

The music dimensions selected in i$^2$MUSE are those frequently used in music searches. These dimensions are also widely included in music applications for everyday activities (e.g., running [13, 14] and driving [15]) as well as therapeutic treatments (e.g., walking training for gait disorders [8, 16]). By adjusting different parameters and weights for each dimension, i$^2$MUSE is able to meet various music information needs and application scenarios.

To use the i$^2$MUSE query interface, users can intuitively click to configure multiple dimensions. Tempo, a real number in beats per minute (BPM), can be input through the text box, via a slider (for a tempo range), or by tapping/mouse clicking (for a single, perceived tempo). Once users set or change the query parameters, they can immediately see comprehensive feedback on the interface, including suggested and un-suggested parameters as well as the result preview (i.e., the percentage of matched tracks in the database). To reduce user operation, an *auto select mode* is also provided. When this mode is switched on, the suggested parameters are automatically selected and included in the final query.

When a user has a clear music information need, or just some vague ideas in mind, he can begin a query from any dimension and then interact with i$^2$MUSE to organize and refine the query. i$^2$MUSE can help bridge user intention gap in serval aspects. First, unlike using text-based music search engines, once the users see their possible choices in multiple music dimensions, they may be prompted to express their intentions more clearly and accurately. Second, i$^2$MUSE allows users to input their intentions more precisely regarding each specific music dimension (e.g., the "happy" category in the mood dimension), which avoids the ambiguity in textual queries (e.g., the word "happy" may match the track title, album, mood or other tags). Third, the interactive query interface and real-time feedback can help users effectively refine their search intentions and construct the queries.

The rest of this paper is organized as follows. Section 2 presents an overview of our system. Section 3 and Section 4 introduce how i$^2$MUSE is constructed and how music search is designed, respectively. Experiments and results are presented in Section 5. Finally, Section 6 concludes our work.

## 2. SYSTEM OVERVIEW

As shown in Figure 2, the framework of i$^2$MUSE consists of two phases: *system construction* and *music search*. In system construction, a database of 12,141 music tracks is built. Each track is analyzed and represented as a music semantic vector (MSV) [2, 7], which is used to index the database. Next, a set of dimension correlation models is learned to support the interactive query interface. In mu-

**Table 1: Six music dimensions used in i²MUSE.**

| Dimension (# Tracks) | Type/Categories (# Tracks) |
|---|---|
| Tempo | Real value between 45~190 |
| Beat strength | Real value between 0~1 |
| Genre (4,285) | Classical (248), Country (356), Electronic (1,285), Hip-hop (170), Jazz (389), Metal (327), Pop (588), and Rock (922) |
| Mood (4,667) | Angry (2,010), Pleasure/Joy (1,509), and Sad (1,048) |
| Instrument (3,121) | Brass (324), Percussion (1,090), Strings (1,117), and Woodwinds (590) |
| Vocal (274) | Female (100), Male (93), Mixed (23), and Non-vocal (58) |

sic search, a user can interact with the query interface to construct a multidimensional music query. Then, the query is matched with each track in the indexed database, and a ranked list of tracks is finally returned according to similarity. Each component of the framework is described in more details in the following.

# 3. SYSTEM CONSTRUCTION

## 3.1 Music Dimensions and Data Collection

To establish the system, we first identify six musical dimensions frequently used in music searches. Among them, two are continuous (tempo and beat strength) and four categorical (genre, mood, instrument and vocal). Tempo measures the average speed of the music in beats per minute (BPM) and beat strength the prominence of beats in the music. Tempo and beat strength are important in various scenarios related to user activities and contexts [8, 13, 14, 15], and the categorical dimensions are among the top music tag groups in describing general musical concepts.

For each categorical dimension, we define several component categories. Specifically, we expand each categorical dimension using multiple lexical resources (e.g., online dictionaries and Google[8]) and ontology/semantic web resources (e.g., WordNet[9] and Wikipedia[10]). A music topology is obtained: each categorical dimension consists of several categories, and each category is related to a number of tags (or keywords), which are then used to build the music database without bias to any dimension.

Two steps are performed to collect the tracks. First, we search for top relevant song titles and artist names using each tag within a category via Last.fm API. Second, each pair of artist name and song title is concatenated as a query to search for music videos on YouTube. For each query, we manually check the top five results and download the audio stream of the one with the highest quality. A total of 12,141 tracks are collected, and the detailed data distribution of each categorical dimension is presented in Table 1.

## 3.2 Content Analysis and Indexing

To support content-based music search using a multidimensional query, each track in the database is represented as a music semantic vector (MSV) for effective indexing. A 21-element MSV is constructed for each track by concatenating all the dimensions and categories in Table 1. The first two elements, tempo and beat strength, are computed directly using Klapuri's algorithm [17] and MIRtoolbox [18, 19], respectively. Each of the rest 19 elements represents the probability that the track belongs to an individual category in the four categorical dimensions, respectively. Because in the initial database (cf. Table 1) each track is only sparsely annotated with one or two categories, machine learning-based classification is used to derive the probability for all categories. Specifically, for each of the four categorical dimensions, a binary SVM classifier [20] is trained for each component category, where the training set consists of all the tracks that have been annotated in that dimension. For example, 8 binary classifiers are trained for the genre dimension using the 4,285 genre-annotated tracks. Each SVM classifier then outputs the probability of the associated category for all tracks (including the training tracks). The overall classification accuracy ($F_1$ score) based on 10-fold cross-validation is 0.56 for Genre, 0.45 for Mood, 0.44 for Instrument and 0.64 for Vocal. These accuracies are acceptable for the subsequent statistical analysis, as demonstrated in [17, 19].

## 3.3 Dimension Correlation Analysis

A set of statistical models of the correlation among different music dimensions is learned to effect the intelligent and interactive query interface. For simplicity, we discretize the continuous tempo and beat strength dimensions evenly into 10 and 5 categories, respectively. The vector representation of each dimension of the $i$-th track is denoted by

$$\mathbf{f}_d^i = [f^i(d,1), f^i(d,2), \cdots, f^i(d,N_d)] , \quad (1)$$

where $d \in \{\text{T,B,G,M,I,V}\}$ represents the dimension of Tempo, Beat strength, Genre, Mood, Instrument, and Vocal, respectively, and $N_d \in \{10, 5, 8, 3, 4, 4\}$ is the corresponding number of categories in each dimension. For $\mathbf{f}_\text{T}^i$ or $\mathbf{f}_\text{B}^i$ of a track, $f^i(d,n) = 1$ and $f^i(d,n') = 0, \forall n' \neq n$ if the tempo or beat strength of the track falls into the $n$-th category of that dimension. For $\mathbf{f}_d^i, d \in \{\text{G,M,I,V}\}$, of a track, each $f^i(d,n)$ is the probability of the $n$-th category given by a binary classifier (cf. Section 3.2).

We assume that given an input query combination of several dimensions, the categories of an un-specified dimension can be modeled by a multivariate Gaussian distribution. Specifically, three cases of query combinations are considered: one dimension where one category is selected (1D1C), one dimension where multiple categories are selected (1DMC), and multiple dimensions in each of which multiple categories are selected (MDMC).

In 1D1C, suppose the $n$-th category of dimension $\delta$ is selected, and the selected category is denoted by $c$. The joint probability of $\delta$ and $c$ conditioned on track $i$ is computed by

$$p(\delta, c \mid i) = \frac{f^i(\delta, n)}{\sum_{\nu=1}^{N_\delta} f^i(\delta, \nu)} , \quad (2)$$

where $N_\delta$ is the number of categories in $\delta$. Given $\delta$ and $c$, we can estimate the resultant Gaussian distribution of an arbitrary un-specified dimension $d$, $\forall d \neq \delta$, by computing its mean ($\boldsymbol{\mu}_{d|\delta,c}$) and covariance ($\boldsymbol{\Sigma}_{d|\delta,c}$) as

$$\boldsymbol{\mu}_{d|\delta,c} = \frac{\sum_i p(\delta, c \mid i)\mathbf{f}_d^i}{\sum_i p(\delta, c \mid i)} , \quad (3)$$

$$\boldsymbol{\Sigma}_{d|\delta,c} = \frac{\sum_i p(\delta, c \mid i)(\mathbf{f}_d^i - \boldsymbol{\mu}_{d|\delta,c})^T(\mathbf{f}_d^i - \boldsymbol{\mu}_{d|\delta,c})}{\sum_i p(\delta, c \mid i)} . \quad (4)$$

[8] https://www.google.com

[9] http://wordnet.princeton.edu

[10] http://www.wikipedia.org

Currently we only use $\boldsymbol{\mu}_{d|\delta,c}$ in the interactive query interface and leave $\boldsymbol{\Sigma}_{d|\delta,c}$ for future study.

Next, 1DMC can be extended from 1D1C. Suppose any $J$ categories of dimension $\delta$ are selected, denoted by $C = \{c_j\}_{j=1}^{J}$. $C$ is treated as a "merged category". The conditional probability of $\delta$ and $C$ on the $i$-th track is given by

$$p(\delta, C \mid i) = \sum\nolimits_{j=1}^{J} p(\delta, c_j \mid i) \,. \tag{5}$$

Then the distribution of an un-specified dimension $d$ (i.e., $\boldsymbol{\mu}_{d|\delta,C}$) can be computed using $\{p(\delta, C \mid i)\}_{\forall i}$ with Eq. 3.

Finally, all input query combinations can be generalized to MDMC. For any $K$ specified dimensions $\Delta = \{\delta_k\}_{k=1}^{K}$, and $\mathbf{C} = \{C_k\}_{k=1}^{K}$, where $C_k$ is the set of selected categories in $\delta_k$, the conditional probability for the $i$-th track is

$$p(\Delta, \mathbf{C} \mid i) = \prod\nolimits_{k=1}^{K} p(\delta_k, C_k \mid i) \,. \tag{6}$$

Accordingly, the mean vector of an un-specified dimension $d$, where $d \notin \Delta$, is computed by

$$\boldsymbol{\mu}_{d|\Delta,\mathbf{C}} = \frac{\sum_i p(\Delta, \mathbf{C} \mid i)\mathbf{f}_d^i}{\sum_i p(\Delta, \mathbf{C} \mid i)} \,. \tag{7}$$

# 4. MUSIC SEARCH

## 4.1 Intelligent and Interactive Query Interface

Based on the dimension correlation analysis, an intelligent and interactive query interface is developed for i$^2$MUSE (cf. Figure 1). As introduced in Section 1, the interface is able to automatically give feedback to a user based on the specified query input. Specifically, given a specified query combination ($\Delta$ and $\mathbf{C}$), the system immediately computes the corresponding $\boldsymbol{\mu}_{d|\Delta,\mathbf{C}}$ (cf. Eq. 7) for each un-specified dimension $d$, $d \notin \Delta$. Then, for each $d$ the interface highlights the category with the largest value in $\boldsymbol{\mu}_{d|\Delta,\mathbf{C}}$ and grays out the one with the smallest value. The highlighted categories may suggest the best combination along with the specified dimensions to meet the user's intention, because these suggested categories are more related to the specific input query. Meanwhile, the un-suggested categories are grayed out to prevent potential conflicts that yields fewer relevant tracks in the output ranked list.

In addition, i$^2$MUSE can provide a real-time illustration of the percentage of relevant tracks in the database in response to a specified query. The percentage is given by

$$R = \frac{\sum_{k=1}^{K} \sum_{j=1}^{J_k} \sum_{i=1}^{Z} f^i(\delta_k, c_{kj})}{\sum_{k=1}^{K} \sum_{n=1}^{N_k} \sum_{i=1}^{Z} f^i(\delta_k, n)} \,, \tag{8}$$

where $J_k$ is the number of selected categories in $\delta_k$, $N_k$ the total number of categories in $\delta_k$, $Z$ the number of tracks in the indexed database, and $c_{kj} \in C_k$.

Finally, users can adjust the relative weight of each specified dimension $\delta \in \Delta$ via an intuitive and convenient slider bar on the query interface.

## 4.2 Query Matching and Ranking

Given a multidimensional query, i$^2$MUSE computes the matching score between the query and the 21-element music semantic vector (MSV) of each track in the database. The MSV of the $i$-th track is denoted as $\mathrm{MSV}_i$, and $\mathrm{MSV}_i(b : e)$ denotes the vector from element $b$ to element $e$. Different types of dimensions have different matching criteria.

For tempo, the query can be input as a single value (e.g., by tapping) or a range (e.g., via the slider). People may perceive the tempo of a song at different metrical levels. For example, given a song with a ground-truth tempo of 80 BPM, some people may perceive 160 or 40 BPM. Therefore, tracks with half or double the query tempo are also given higher priorities. When handling a single query tempo $q_{\mathrm{T}}$, the matching score of track $i$ is defined as

$$\begin{aligned} S(q_{\mathrm{T}}, i) = {} & w_0 \cdot \phi\left(q_{\mathrm{T}}/2, \mathrm{MSV}_i(1)\right) + \\ & w_1 \cdot \phi\left(q_{\mathrm{T}}, \mathrm{MSV}_i(1)\right) + w_2 \cdot \phi\left(2q_{\mathrm{T}}, \mathrm{MSV}_i(1)\right) \,, \end{aligned} \tag{9}$$

where $\phi(\cdot,\cdot)$ is a radial basis function (RBF) between two values, $\mathrm{MSV}_i(1)$ the tempo (i.e., first element of MSV) of track $i$, $w_1$ the weight for $q_{\mathrm{T}}$, and $w_0$ and $w_2$ the weights for half and double of $q_{\mathrm{T}}$, respectively. Given a query tempo range $[q_{\mathrm{T}1}, q_{\mathrm{T}2}]$, the matching score is measured by

$$S'(q_{\mathrm{T}1}, q_{\mathrm{T}2}, i) = \begin{cases} S(q_{\mathrm{T}1}, i) & \mathrm{MSV}_i(1) < q_{\mathrm{T}1} \,, \\ S(q_{\mathrm{T}2}, i) & \mathrm{MSV}_i(1) > q_{\mathrm{T}2} \,, \\ 1 & \text{otherwise} \,. \end{cases} \tag{10}$$

For a query beat strength $q_{\mathrm{B}}$, the matching score is computed as $-D(q_{\mathrm{B}}, \mathrm{MSV}_i(2))$, where $D(\cdot,\cdot)$ is the Euclidean distance. As for a categorical music dimension $d$, the query is represented as a vector $\mathbf{f}_d^q$, where $d \in \{\mathrm{G}, \mathrm{M}, \mathrm{I}, \mathrm{V}\}$ and $f^q(d, n)$ is 1 if the $n$-th category is selected, or 0 otherwise. The matching score is defined as $-D(\mathbf{f}_d^q, \mathrm{MSV}_i(b_d : e_d))$, where $b_d$ and $e_d$ are the beginning and ending elements of dimension $d$, respectively.

As i$^2$MUSE allows users to set the relative weight of each specified dimension in the query, the final matching score of a track is the weighted combination of matching scores of the specified dimensions. The ranked list of tracks is then generated by sorting the final scores in a decreasing order.

# 5. EVALUATION AND RESULTS

## 5.1 System Evaluation

### 5.1.1 *Configuration*

Due to the difficulty of directly comparing MMSEs (which have fixed categories and dimensions) with free-text based search engines, and existing studies that have demonstrated the advantages of MMSEs (cf. [8, 9]; also described in Section 1), in this paper we only implemented and compared our proposed i$^2$MUSE with another baseline MMSE. This baseline MMSE had the same music dimensions and categories as i$^2$MUSE but without those feedback functions, including automatic category suggestion/un-suggestion, weight adjustment, and result preview. Both i$^2$MUSE and the baseline used the same music database (cf. Section 3.1) and indexing and query matching algorithms (cf. Section 3.2 and 4.2).

We designed two modes for subjects to conduct multidimensional music search. In the *search-by-scenario* mode, a subject was given a real-life scenario (e.g., Running in the morning) and then searched for music appropriate for it. In the *search-by-example* mode, a subject listened to an example song and then searched for music which sounded similar. We defined 10 scenarios (Table 2) covering different music dimensions and categories. Similarly, 10 example songs with different styles were also selected from the database.

Table 2: Ten real-life scenarios.

| No. | Scenario Description |
|---|---|
| 1 | Running in the morning |
| 2 | Dancing at the club |
| 3 | Relaxing in bed |
| 4 | Reading in the library |
| 5 | Shopping in a supermarket |
| 6 | Alone on a rainy day |
| 7 | Watching boxing |
| 8 | On the way back home |
| 9 | On the bus |
| 10 | Sports meet |

### 5.1.2 Experimental Methodology

We recruited 30 subjects with experience of online music search for this evaluation. After a brief introduction of the evaluation interface, each subject was asked to perform 20 search trials (10 by scenarios and 10 by example songs) using both search engines, with 10 trials (5 by scenarios and 5 by songs) for each search engine (i.e., no repeated trials for each subject). Both search engines were evaluated with all 10 scenarios and 10 example songs by an equal number of 15 subjects.

For each trial, the input query and the result list were saved for further analysis. User operations (e.g., clicking) during search were also recorded. After finishing all trials, the usability of each feedback function in i$^2$MUSE (e.g., suggested category) was assessed by users. The overall system usability of i$^2$MUSE was also evaluated through a questionnaire. A five-point score was adopted for these ratings.

## 5.2 Effectiveness Study

To evaluate the effectiveness of the search engines, we can investigate, in the search-by-example mode, how well an example song was ranked, as measured by the mean reciprocal rank (MRR) of the example song. As shown in Figure 3(a), i$^2$MUSE achieved higher MRR values than the baseline MMSE, with average MRR significantly improved by 35% ($p$-value<0.05). This indicates that i$^2$MUSE ranked the example song (that the subjects intended to search) higher in the result list than the baseline.

As introduced in Section 5.1.1, the only difference between these two systems is that i$^2$MUSE provides multiple feedback functions (e.g., suggested category), while the baseline does not. Therefore, the improved retrieval performance of i$^2$MUSE shows that multiple feedback functions can indeed help users construct queries that express their search intentions more accurately.

## 5.3 Usability Study

The usability of the i$^2$MUSE system is closely related to the performance and user acceptance of each feedback function. We calculated the average usability ratings of four feedback functions and the overall user interface (UI) effects of i$^2$MUSE. As shown in Table 3, the subjects were satisfied with the suggested category, weight adjustment and UI effects. While the ratings for the un-suggested category function was lower, it remained favorable.

We further analyzed the collected logs of user operations to see whether and how often the subjects had adopted the feedback functions. Figure 3(b) and Figure 3(c) present the adoption rates of both suggested category and un-suggested category in the search-by-scenario and search-by-example modes, respectively. For suggested category, the higher the adoption rate is the more useful it is, whereas for un-suggested category, the lower the better.

Table 3: Usability ratings on i$^2$MUSE feedback functions. Scale: 1 (very dissatisfied) – 5 (very satisfied).

| Feedback Functions | Ratings |
|---|---|
| suggested category | 4.07 |
| un-suggested category | 3.23 |
| result preview | 3.40 |
| weight adjustment | 3.73 |
| UI effects | 4.07 |

Table 4: Questionnaire for overall usability study.

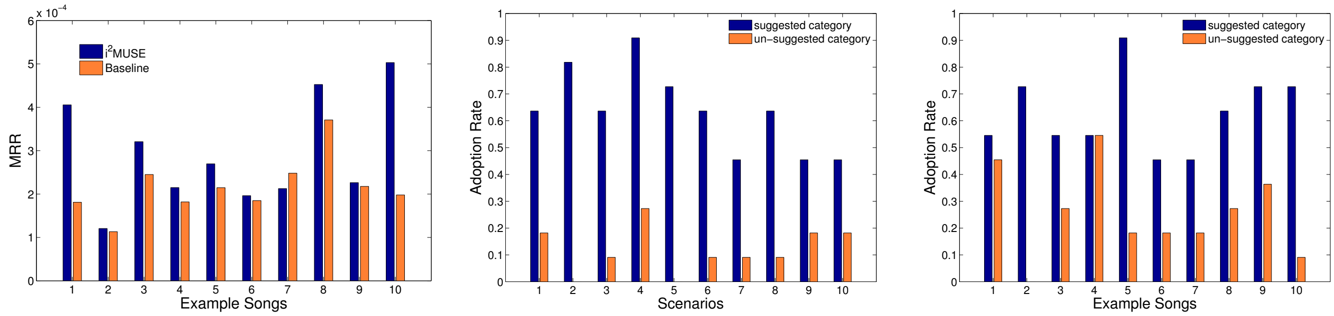
| No. | Question Description |
|---|---|
| 1 | I am satisfied with the overall system. |
| 2 | I can easily search what I want. |
| 3 | Search interface is easy to understand. |
| 4 | I am satisfied with steps included for a search task. |
| 5 | I need additional instructions to complete the search. |
| 6 | There is information overload on webpage. |

For all 20 search trials, the average adoption rates for the above two functions were 0.61 and 0.13, respectively. These results confirmed the efficacy of both feedback functions. Suggested categories were adopted by users in most of search trials. Un-suggested categories were also selected in some cases (e.g., example song 1 and 4). Especially, in the search-by-song mode, un-suggested categories were adopted more often than those in the search-by-scenario mode. This was because when given a scenario, users could still be vague and subjective towards what they wanted to find, while when given an example song their search intentions became more clear and specific. Therefore, categories that were un-suggested based on our statistical model might be not very accurate for some search-by-song cases.

To obtain an overall usability evaluation from the users, a questionnaire survey of 6 questions was conducted (cf. Table 4), and each question was rated on a five-point scale. The survey results showed that, firstly, users were satisfied with the general usability (No.1: 3.7) and could easily find what they wanted (No.2: 3.4). Moreover, the interaction and UI design were friendly to the users with high ratings (No.3: 4.0 and No.4: 4.1). Finally, despite the fact that more functions and steps were introduced in i$^2$MUSE, the interface was easy to understand and no more cumbersome to users, as question No.5 and No.6 both had low ratings (2.2 and 2.0 respectively). In conclusion, the survey results positively confirmed both the effectiveness and usability of i$^2$MUSE.

# 6. CONCLUSION AND FUTURE WORK

In this paper, we have presented i$^2$MUSE, a novel intelligent and interactive multidimensional music search engine. Six content-based dimensions (tempo, beat strength, genre, mood, instrument, and vocal) are incorporated in i$^2$MUSE. On the interactive query interface, users can organize and input music queries with multiple dimensions efficiently and effectively. Once some dimensions have been specified by the user, i$^2$MUSE is able to instantaneously highlight a suggested parameter and gray out an un-suggested parameter for every other dimension. Simultaneously, the percentage of matched tracks in the database is illustrated on the inter-




(a) Mean Reciprocal Ranks of 10 example songs in the search-by-example mode

(b) i$^2$MUSE suggestion function adoption rates (search-by-scenario mode).

(c) i$^2$MUSE suggestion function adoption rates (search-by-example mode).

**Figure 3: Results of i$^2$MUSE effectiveness and usability study.**

face. Users can also adjust the weight of each specified dimension. The interactive and intelligent features of i$^2$MUSE could help users express their music intentions more intuitively and clearly, and potentially prevent parameter conflicts in the query. A pilot user study has been conducted and validated the effectiveness and usability of i$^2$MUSE.

At present, our system is relatively limited in track scale and the number of dimensions and their categories. For future work, we plan to expand the database and further explore different dimension correlation analysis methods. We could also combine the statistical correlation model with collaborative filtering or user profile/search history to further refine user intention in the future.

## ACKNOWLEDGEMENTS

We thank Shant Sagar for his help in the query interface design. This project is funded by the National Research Foundation (NRF) and managed through the multi-agency Interactive & Digital Media Programme Office (IDMPO) hosted by the Media Development Authority of Singapore (MDA) under Centre of Social Media Innovations for Communities (COSMIC).

## 7. REFERENCES

[1] R. Typke, F. Wiering, and R. C. Veltkamp. A survey of music information retrieval systems. In *Proc. of ISMIR*, pages 153–160, 2005.

[2] Z. Li, B. Zhang, et al. Query-document-dependent fusion: A case study of multimodal music retrieval. *IEEE Trans. on Multimedia*, 15(8):1830–1842, 2013.

[3] Y. Yu, R. Zimmermann, Y. Wang, and V. Oria. Scalable content-based music retrieval using chord progression histogram and tree-structure LSH. *IEEE Trans. on Multimedia*, 15(8):1969–1981, 2013.

[4] Z. J. Zha, L. Yang, T. Mei, et al. Visual query suggestion: Towards capturing user intent in internet image search. *ACM Trans. Multimedia Comput. Commun. Appl.*, 6(3):13:1–13:19, August 2010.

[5] A. Hanjalic, C. Kofler, and M. Larson. Intent and its discontents: the user at the wheel of the online video search engine. In *Proc. of ACM Multimedia*, pages 1239–1248, 2012.

[6] Peeters G., Cornu F., et al. A multimedia search and navigation prototype, including music and video-clips. In *Proc. of ISMIR*, pages 439–444, 2012.

[7] B. Zhang, J. Shen, Q. Xiang, et al. CompositeMap: A novel framework for music similarity measure. In *Proc. of ACM SIGIR*, pages 403–410, 2009.

[8] Z. Li, Q. Xiang, J. Hockman, et al. A music search engine for therapeutic gait training. In *Proc. of ACM Multimedia*, pages 627–630, 2010.

[9] A. Lenoir, R. Landais, and J. Law-To. MuMa: A scalable music search engine based on content analysis. In *Proc. of CBMI*, pages 1–4, 2012.

[10] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proc. of ACM SIGIR*, pages 4–11, 1996.

[11] B. Vélez, R. Weiss, M. A. Sheldon, et al. Fast and effective query refinement. In *ACM SIGIR Forum*, volume 31, pages 6–15, 1997.

[12] H. Cao, , D. Jiang, J. Pei, et al. Context-aware query suggestion by mining click-through and session data. In *Proc. of ACM SIGKDD*, pages 875–883, 2008.

[13] S. Nirjon, R. F. Dickerson, Q. Li, et al. Musicalheart: A hearty way of listening to music. In *Proc. of ACM SenSys*, pages 43–56, 2012.

[14] B. Moens, L. van Noorden, and M. Leman. D-jogger: Syncing music with walking. In *7th SMC Conference*, pages 451–456. Universidad Pompeu Fabra, 2010.

[15] L. Baltrunas, M. Kaminskas, B. Ludwig, et al. Incarmusic: Context-aware music recommendations in a car. In *E-Commerce and Web Technologies*, pages 89–100. Springer, 2011.

[16] S. Zhu, R. J. Ellis, et al. Validating an iOS-based rhythmic auditory cueing evaluation (iRACE) for Parkinson's disease. In *Proc. of ACM Multimedia*, 2014. http://dx.doi.org/10.1145/2647868.2654952.

[17] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the meter of acoustic musical signals. *IEEE TASLP*, 14(1):342–355, 2006.

[18] O. Lartillot and P. Toiviainen. MIR in Matlab (II): A toolbox for musical feature extraction from audio. In *Proc. of ISMIR*, pages 127–130, 2007.

[19] O. Lartillot, T. Eerola, P. Toiviainen, et al. Multi-feature modeling of pulse clarity: Design, validation, and optimization. In *Proc. of ISMIR*, pages 521–526, 2008.

[20] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2:1–27, 2011.