PAPER

Robust and Fast Phonetic String Matching Method for Lyric Searching Based on Acoustic Distance

Xin XU[†a)], ***Member and*** **Tsuneo KATO**[†], ***Senior Member***

SUMMARY This paper proposes a robust and fast lyric search method for music information retrieval (MIR). The effectiveness of lyric search systems based on full-text retrieval engines or web search engines is highly compromised when the queries of lyric phrases contain incorrect parts due to mishearing. To improve the robustness of the system, the authors introduce acoustic distance, which is computed based on a confusion matrix of an automatic speech recognition experiment, into Dynamic-Programming (DP)-based phonetic string matching to identify the songs that the misheard lyric phrases refer to. An evaluation experiment verified that the search accuracy is increased by 4.4% compared with the conventional method. Furthermore, in this paper a two-pass search algorithm is proposed to realize real-time execution. The algorithm pre-selects the probable candidates using a rapid index-based search in the first pass and executes a DP-based search process with an adaptive termination strategy in the second pass. Experimental results show that the proposed search method reduced processing time by more than 86.2% compared with the conventional methods for the same search accuracy.

key words: *lyric search, phonetic confusion matrix, two-pass search, dynamic programming*

1. Introduction

Current commercial music information retrieval (MIR) systems accept queries in a range of forms by text, humming, singing, and acoustic music signals. Among these, text queries of lyric phrases are commonly used [1]. As many MIR systems apply full text search engines to lyric search, it has been widely regarded that the issue of lyric search has been solved by state-of-the-art text retrieval techniques. However, there is a problem. The investigations on real world queries, as conducted in this paper, suggested that users are likely to input incorrect lyric phrases into MIR systems, resulting in a failure. The investigation found that incorrect queries that replace a word with another word of a similar pronunciation can occur at rates as high as 19.3% of total collected lyric queries. These incorrect lyric queries are due to mishearing or the unreliability of human memory, as users only memorize the lyric phrases when they enjoy hearing a part of a song and usually do not use the aid of its lyric sheet. The investigation also verified that major commercial web search engines implemented with fuzzy matching algorithms were not helpful. A search method is expected to be able to identify the lyric containing the part that is most acoustically similar to the query. Phonetic string matching, which is used in such applications as name retrieval [2], was considered to be an appropriate method to solve this problem of low search accuracy.

However, another important requirement for lyric search is that it must satisfy a real-time response. As the search algorithm of phonetic string matching methods is based on exhaustive dynamic programming (DP), the computational complexity results are in the order of $m * n * I_t$ per query. Here m is the length of the query, n is the average length of a lyric and I_t is the number of lyrics to search. Since commercial MIR systems usually provide hundreds of thousands of lyrics, the computational complexity is too high to realize a real-time search.

In order to solve these two problems peculiar to lyric search, in this paper the authors propose a novel method to make lyric search simultaneously robust and fast.

First, in order to improve lyric search accuracy, the proposed method applied "acoustic distance" to measure the acoustic confusability between phonetic strings due to mishearing. The acoustic distance values are derived from a DP-based phonetic string matching calculation using a phoneme confusion matrix to quantize acoustic similarity between phonemes. The values in the confusion matrix are obtained by an automatic speech recognition (ASR) experiment. Although the confusion matrix should be based on singing voice, a huge amount of singing data is not available. In this paper, telephone speech data of Japanese phonetically balanced sentences, which are more easily-obtainable data in the field of speech recognition, were used as the training data for the acoustic models of ASR. This is inspired by concepts in Spoken Document Retrieval (SDR) and Spoken Utterance Retrieval (SUR) [3], [4].

To solve the second problem in real-time search, a two-pass search algorithm is applied in the proposed lyric search method. It uses a fast inverted-index-based search in the first pass and DP-based search with an adaptive termination strategy in the second pass. In the first pass, the proposed method pre-selects the probable lyric candidates by means of a rapid approximate search based on the accumulation of pre-computed and indexed partial acoustic distances. Then, in the second pass, the lyric candidates are sorted by the approximate acoustic distances and evenly divided into groups. The exhaustive DP matching between the query and the lyrics is carried out group by group. During the DP matching, a cut-off function for the adaptive termination is calculated by the DP distances to make the matching process more efficient. Once the function value exceeds a predeter-

Manuscript received November 18, 2013.
Manuscript revised May 7, 2014.

[†]The authors are with the KDDI R&D Laboratories Inc., Fujimino-shi, 356–8502 Japan.
a) E-mail: sh-jo@kddilabs.jp

DOI: 10.1587/transinf.2013EDP7418

Copyright © 2014 The Institute of Electronics, Information and Communication Engineers

mined threshold for some group, which means the correct lyric has been found, the search is terminated. The experimental results show that the processing time is greatly reduced by using the proposed two-pass search strategy, without loss of search accuracy. The group size that contributed to the best performance was proved to be 100 lyrics, which is about one fifteenth of the candidates.

The remainder of this paper is organized as follows: Section 2 presents related works on lyric search. The analysis of mistaken queries is described in Sect. 3. Section 4 introduces how to calculate acoustic distance during phonetic string matching. All of the search processes of the proposed lyric search method are described in detail in Sect. 5. In Sect. 6, the experiments are carried out to evaluate the proposed method in terms of search accuracy and processing time. The paper is summarized in Sect. 7.

2. Related Works

Several related studies attempted to use phonetic string matching methods to solve the search problem caused by misheard lyrics. They were verified to be more robust than the text retrieval methods. Ring and Uitenbogerd [5] tried to find the correct lyric by minimizing the edit distances between phoneme strings of queries and the lyrics. However, edit distance does not present the degree of confusability between phonemes. To model the similarity of misheard lyrics to their correct versions statistically, Hussein [6] introduced a probabilistic model of mishearing that is trained using examples of actual misheard lyrics from a user-submitted misheard lyrics website “kissthisguy” [7], and developed a phoneme similarity scoring matrix based on the model. The performance of this method depends on the size of the training database. As described in [6], a total number of 20788 pairs of the misheard lyrics and the correct lyrics are used. However, such a big database like “kissthisguy” is not available in other languages. For example, in order to search lyircs in Japanese, it is impractical to collect sufficient misheard lyrics to build a practical probabilistic model.

On the other hand, in order to reduce the processing time, conventional high-speed DP matching processors use index or tree-structured data to pre-select the hypothetical candidates [8], [9]. As an example, [8] used a suffix array as the data structure and applied phoneme-based DP matching to detect keywords quickly from a very large speech database. In order to avoid an exponential increase in the processing time caused by increasing keyword length, it divided the original keyword into short sub-keywords. Then, it searched the sub-keywords on the suffix array by DP matching. If the DP distance between a sub-keyword and a path of the suffix array is not more than a predetermined threshold value, these pathes remained as the candidates of search results. By repeating the DP matching process between the original keyword and the candidates, the final result is detected. As well as other high-speed DP methods, the predetermined threshold for sub-keywords is proportional to the length of the queries.

However, lyric search has a distinctive characteristic: it is too difficult to determine an absolute threshold to decide whether a lyric is the exactly correct one for the incorrect query or not, since it is related to the individual variations of mishearing. Therefore, the previous studies on lyric search used the common criterion of looking up the entire lyric search space and estimating the lyric at minimum distance from the query to be the user’s target. Based on the investigation of real world queries in Sect. 3.3 of this paper, the DP distances between the queries and the correct lyrics have no statistical relationship with the lengths of the queries. The conventional high-speed DP processors are not able to keep high search accuracy for the lyric search case.

3. Analysis of Real World Lyric Queries

Several investigations were carried out on collected real world queries. The statistical features of queries and some issues peculiar to lyric search are presented in this section.

3.1 Statistical Features of Real World Queries

To analyze the queries of lyric phrases for MIR in the real world, the authors investigated major Japanese question & answer community websites, “okwave” [10] and “oshiete goo” [11]. It was found that many questions used lyric phrases to request the names of songs and singers. As 1140 queries of lyric phrases asked by various questioners were collected, the authors compared each query with its corresponding lyric to categorize whether lyric phrases in the query are correct or not (correct query or incorrect query) and how they were mistaken. The lyrics and queries are written in Japanese or English, or a mixture of both.

Figure 1 shows the distribution of incorrect queries in the different types and correct queries within the collected data. The incorrect queries, which make up around 79%, are classified into the following types:

- Confusion of notations: Chinese characters in the queries are substituted for reading symbols (kana), and

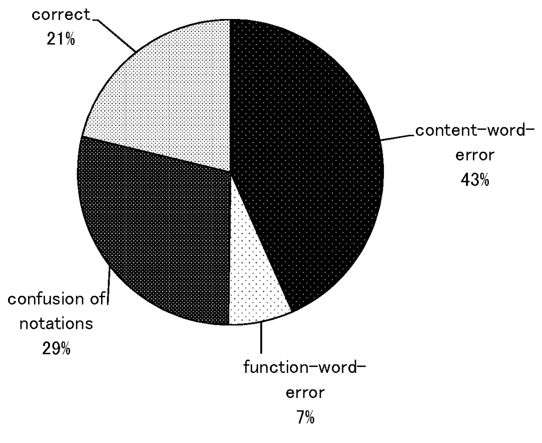

Fig. 1 The distribution of mistaken queries in the different types and correct queries within the collected queries.

Table 1 The distribution of mistaken types within content-word-error cases.

Types of queries	Percentage			Examples	
				Correct lyric	Mistaken queries
Acoustic confusion	19.3% (220 queries)	Ex.1	Text (Japanese):	好きな**事がない**	好きな**言葉は何**
			Pronunciation:	/sukina**kotoganai**/	/sukina**kotobawanani** /
			Meaning:	There is **nothing** I like.	What are your favorite **words**?
		Ex.2	Text:	**You've been out riding fences for so long now**	**ユーベーナウプラウドゥンシーンクスソーセングナウ**
			Pronunciation:	**/yuubiibiiNNautoraidiNNgufeNNshizufoosooroNNgunau /**	**/yuubeenaupurauduNNshiiNNkususooseNNgunau/**
			Meaning:	You've been out riding fences for so long now	* no actual meaning
Meaning confusion	7.3% (83 queries)	Ex.1	Text (Japanese):	**君**には何でも話せるよと	**あなた**には何でも話せるよと
			Pronunciation:	/**kimi**niwanaNNdemohanaseruyoto/	/**anata**niwanaNNdemohanaseruyoto/
			Meaning:	I can say anything to **you**	I can say anything to **you**
		Ex.2	Text (Japanese):	**月**に願いを	**星**に願いを
			Pronunciation:	/**tsuki**ninegaio/	/**hoshi**ninegaio/
			Meaning:	pray to the **moon**	pray to the **star**
Others	16.3% (186 queries)	Ex.1	Text (Japanese):	星**から来た**子の見る夢は	星の子**チョビン**の見る夢は
			Pronunciation:	/hoshi**karakita**konomiruyumewa /	/hoshinoko**chobiNN**nomiruyumewa /
			Meaning:	The dream that the child who **came from** the star has	The dream that child **Chobin** of the star has

vice versa.

- Function-word-error: Only the function words (such as prepositions, pronouns, auxiliary verbs), which have little lexical meaning, are mistaken in the queries.
- Content-word-error: The content words (such as nouns, verbs, or adjectives), which have stable lexical meanings, are mistaken in the queries.

In the current full-text search methods, function-word-error and confusion of notations can be handled using a stop word list for filtering out the function words [12], and a hybrid index of words and syllables [13].

On the other hand, as the content words play more important roles in determining the search intention [12], content-word-error queries were further categorized into three subtypes by the authors, namely "acoustic confusion", "meaning confusion" and "others". The percentages and examples are listed in Table 1. The mistaken parts are marked in bold.

Acoustic confusion is defined as a replacement of a word with that of a similar pronunciation; or a replacement of the words of unknown spelling with reading symbol strings of a similar pronunciation. For the first example of acoustic confusion queries in Table 1, "/kotoganai/" and "/kotobawanani/" have similar pronunciations while the character strings have no common parts. In the second example, the Japanese syllable (kana) string is used as a query whose pronunciation is similar to the English phrase, "You've been out riding fences for so long now" in the target lyric. This was assumed to happen when users were not able to spell the foreign words that they heard in a song.

Meaning confusion is defined as a replacement of a word with its synonym or near-synonym. As shown in Table 1, in the first example of meaning confusion queries, "/anata/" is mistaken for "/kimi/". Both of the terms refer to the same meaning "you" in Japanese. For the second example, "/tsuki/" and "/hoshi/", which mean "moon" and "star", are confused.

The "others" type contains word insertion, word deletion and other errors in the queries. From the analysis of collected examples, it is known that mistakes in the "others" type are caused by a variety of reasons, which include individual experiences or memories and other reasons. The analysis did not find a relationship between the mistakes and the lyrics.

As the acoustic confusion queries occupy about 19.3% of the collected queries (45.0% of content-word-error queries), it remains an important issue for lyric search.

3.2 Search Test of Acoustic Confusion Queries by Web Search Engines

This paper focuses on the solution to the acoustic confusion issue for lyric search. The average length of 220 acoustic

Table 2 Number of hits by two web search engines.

web search engines	Web Search Engine 1	Web Search Engine 2
220 correct queries	175	157
220 incorrect queries	27	16

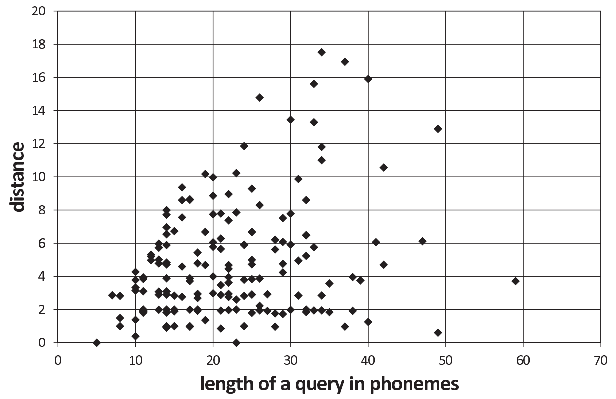

Fig. 2 The distribution of the length of a query in phonemes and the DP matching distances from the correct lyric for the real world incorrect queries.

confusion queries collected is about 6 words. The word error rate of incorrect words is about 53.1% (the insertion errors were not included). In the text retrieval field, some fuzzy matching algorithms, such as Latent Semantic Indexing (LSI) and partial matching, were used by major commercial Web search engines [14] to improve the robustness against incorrect queries. Thereby, a search test was carried out to evaluate how robust web search engines are against 220 acoustic confusion queries collected. The test results are shown in Table 2. The number of "hits" is equal to the number of webpages mentioning the target lyric that are included in the top 20 results returned by a search engine. Correct queries mean the correct versions of the incorrect queries. Comparing the number of hits with the correct queries, the performance of both web search engines are severely degraded in case of incorrect queries.

According to this result, identifying a lyric containing the most similar part in the acoustic aspect of the query is expected to be a better solution for acoustic confusion than focusing on the textual or the semantic aspects.

3.3 Investigation of the Relationship between the Lyric Query Length and DP Matching Pre-Selection

As introduced in Sect. 2, the conventional high-speed DP-based search method usually contains a pre-selection approach. It prunes out the improbable search paths by compareing the DP matching distance with the predetermined threshold proportional to the length of queries. To find out whether it is a practicable approach to the lyric search problem, the DP matching distances between the queries and the correct lyric are also analyzed. Figure 2 shows the distribution of the analysis data. The horizontal axis is the phoneme number of each query, representing the length of the queries. The vertical axis is the DP matching distance between the queries and the correct lyrics. Figure 2 shows that the phoneme number of the queries is distributed in a broad range from 5 to 57. In addition, the distance values between the queries and the correct lyrics show no statistical relationship with the length of queries. Thereby, it is practically difficult for the conventional method, such as the one in [8], to find the appropriate threshold based solely on the length of queries.

4. Acoustic Distance Derived from a Phoneme Confusion Matrix

In this section, the proposed acoustic distance is presented. Acoustic distance between two strings is calculated by DP matching with cost values derived from phonetic confusion probabilities instead of a constant cost value used for edit distance.

First, a phonetic confusion matrix is obtained by running a phoneme speech recognizer over a set of speech data and aligning the phoneme strings of recognition results with reference phoneme strings, which uses the same speech recognition experiment as in [15].

For the elements of the confusion matrix, $g(p, q)$ means the number of instances of phoneme q obtained as recognition results by the actual utterances of phoneme p. As "ϕ" represents a null, $g(\phi, p)$ means the number of instances of the wrongly inserted phoneme p (insertion) and $g(p, \phi)$ means the number of instances of the deleted phoneme p (deletion). U represents the set of 37 phonemes including null.

For each phoneme p, the phonetic confusion probabilities of an insertion $P_{ins}(p)$, deletion $P_{del}(p)$ and substitution for phoneme q $P_{sub}(p, q)$ are calculated on the basis of the confusion matrix elements, by Eq. (1)~(3).

$$P_{ins}(p) = \frac{g(\phi, p)}{\sum_{k \in U} g(k, p)} \tag{1}$$

$$P_{del}(p) = \frac{g(p, \phi)}{\sum_{k \in U} g(p, k)} \tag{2}$$

$$P_{sub}(p, q) = \frac{g(p, q)}{\sum_{k \in U} g(p, k)} \tag{3}$$

As a large value of $P_{ins}(p)$ represents high confusability for an insertion of p, it corresponds to the low cost of an insertion operation for p in string matching based on DP. Therefore the value of insertion cost $C_{ins}(p)$ is calculated by Eq. (4). In the same way, the value of deletion cost $C_{del}(p)$ and substitution cost $C_{sub}(p, q)$ are calculated from the corresponding phonetic confusion probabilities by Eq. (5) and Eq. (6).

$$C_{ins}(p) = 1 - P_{ins}(p) \tag{4}$$

$$C_{del}(p) = 1 - P_{del}(p) \tag{5}$$

$$C_{sub}(p, q) = 1 - P_{sub}(p, q) \tag{6}$$

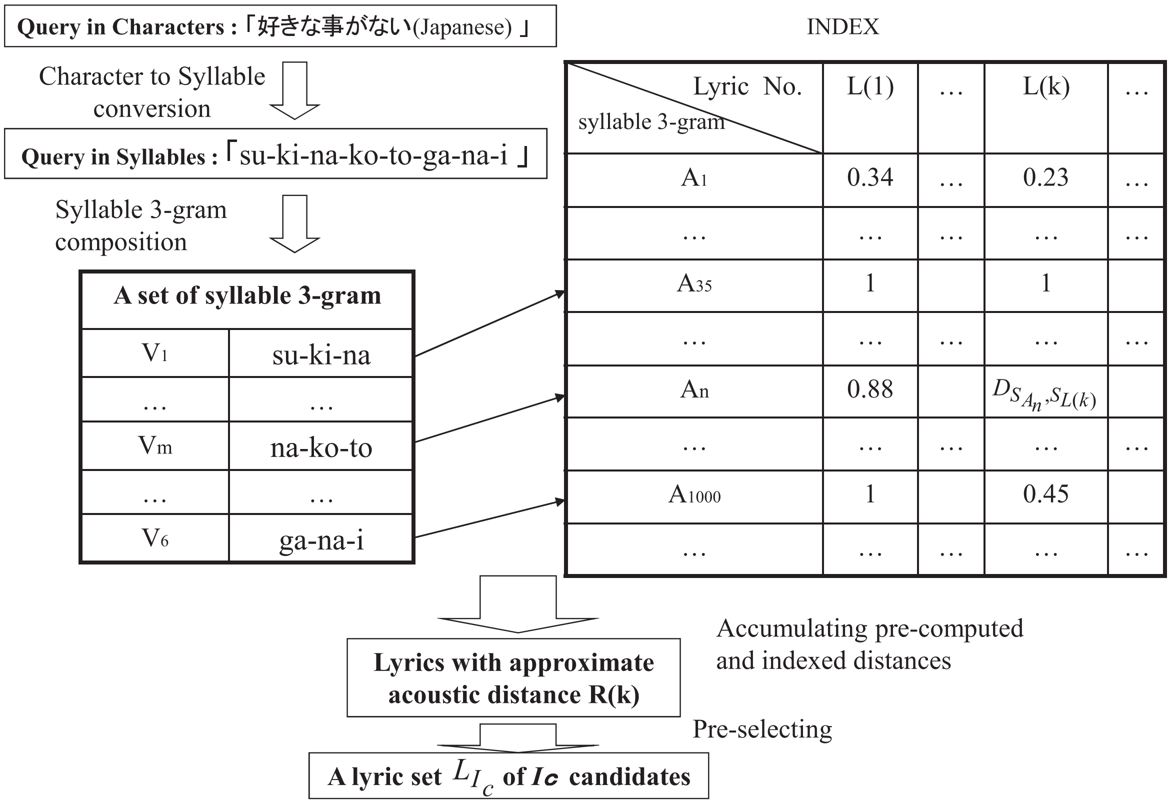
Query in Characters :「好きな事がない(Japanese)」
Character to Syllable conversion
Query in Syllables :「su-ki-na-ko-to-ga-na-i」
Syllable 3-gram composition
A set of syllable 3-gram
V1 su-ki-na
Vm na-ko-to
V6 ga-na-i
INDEX
Lyric No.
syllable 3-gram
L(1)
L(k)
A1 0.34 0.23
A35 1 1
An 0.88 $D_{S_{A_n},S_{L(k)}}$
A1000 1 0.45
Accumulating pre-computed and indexed distances
Lyrics with approximate acoustic distance R(k)
Pre-selecting
A lyric set L_{I_c} of I_c candidates

Fig. 3 Flowchart of the first pass search.

Second, with the calculated cost values, edge-free DP matching between the phoneme strings S_1, S_2 is carried out by Eq. (7)~(9). Here, $S[x]$ is xth phoneme of phoneme string S and $len(S)$ means the length of S ($S_1, S_2 \in S$). $D(i, j)$ designates the minimum distance from the starting point to the lattice point (i, j). D_{S_1,S_2} is the accumulated cost of DP matching between S_1 and S_2, which is defined as the acoustic distance. It reflects acoustic confusion probability for each phoneme.

1. Initialization:

$$D(0, j) = 0 \ (0 \le j \le len(S_2)); \tag{7}$$

2. Transition:

$$D(i, j) = \min \begin{cases} D(i, j-1) + C_{ins}(S_2[j]) \\ D(i-1, j-1) + C_{sub}(S_1[i], S_2[j]) \\ D(i-1, j-1), (if\ S_1[i] = S_2[j]) \\ D(i-1, j) + C_{del}(S_1[i]) \end{cases} \tag{8}$$

3. Determination

$$D_{S_1,S_2} = min\{D(len(S_1), j)\} \ (0 < j \le len(S_2)); \tag{9}$$

5. Fast Two-Pass Search Algorithm in Consideration of Acoustic Similarity

A two-pass search strategy is proposed to be used in the DP-based phonetic string matching, which is based on acoustic distance, in order to realize a real-time search. It is realized through the following steps: off-line index construction, a rapid index-based search in the first pass and a DP-based search process with an adaptive termination strategy in the second pass.

5.1 Preliminary Indexing

Theoretically, DP matching computation for the acoustic confusion distance between queries and lyric text should be done beforehand. However, this is impossible in reality because the number of query patterns is too large to be predicted.

An inverted index construction is preliminarily incorporated for the first pass search. The whole lyric set L_{I_t} are converted into syllable strings using a morphological analysis tool such as Mecab [16]. Here I_t represents the number of lyrics in the whole set. The syllable strings are converted into phoneme strings by referring to a syllable-to-phoneme translation table. Consequently, a phoneme string $S_{L(k)}$ represents a lyric $L(k)$ ($L(k) \in L_{I_t}$). Here k is the lyric number. On the other hand, a list of linguistically existing units of N successive syllables (syllable N-gram) $A_1 \cdots A_n$ are collected from the lyric corpus. The units are organized as index units for fast access, as shown in Fig. 3. The acoustic distance $D_{S_{A_n},S_{L(k)}}$ between the phoneme strings of A_n and $L(k)$ are pre-computed by Eq. (7)~(9) and stored in the index matrix. It can be regarded as an index of acoustic confusion.

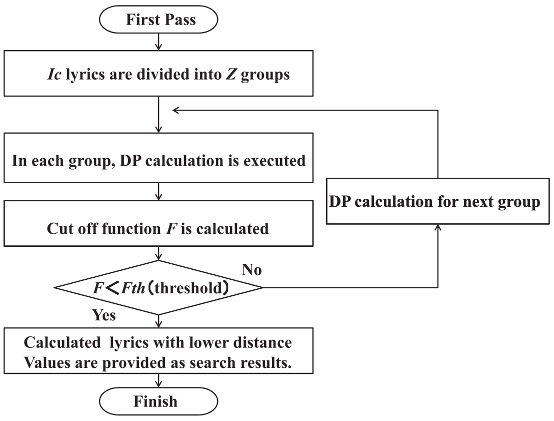
First Pass
Ic lyrics are divided into *Z* groups
In each group, DP calculation is executed
Cut off function *F* is calculated
DP calculation for next group
F<*Fth* (threshold)
No
Yes
Calculated lyrics with lower distance Values are provided as search results.
Finish

Fig. 4 Flowchart of the second pass search.

5.2 Index Search in the First Pass

By accessing the index described above, a fast search is realized using the following steps. The flowchart is shown in Fig. 3:

1. The input query Q is converted into a syllable string v by Macab.
2. By Eq. (10) the syllable string is converted into syllable N-gram sets, $V_1, \ldots, V_m, \ldots, V_M$. Here, $v[m]$ is the mth syllable of v.

$$V_m = \{v[m], v[m+1], \cdots, v[m+N-1]\}; \qquad (10)$$

3. $V_1, \ldots, V_m, \ldots, V_M$ are matched with the index units $A_1, \ldots, A_n, \ldots$. By accumulating the pre-computed and indexed distance values $D_{S_{A_n}, S_{L(k)}}$, the approximate acoustic distance $R(k)$ is calculated by Eq. (11).

$$R(k) = \sum_{m=1}^{M} D_{S_{A_n=V_m}, S_{L(k)}} \qquad (11)$$

4. To narrow the search space of lyrics, $L(k)$ with higher $R(k)$ is pruned off, and a lyric set L_{I_c} containing I_c ($I_c < I_t$) as the best lyric candidates is preserved for the second pass.

As seen in the four steps, the order of the syllable N-grams is not considered in the first pass.

5.3 DP-Based Search Process with an Adaptive Termination Strategy in the Second Pass

By means of the pre-selection in the first pass, the range of target lyrics is narrowed down to L_{I_c}. DP matching with the lyrics in L_{I_c} is then carried out to calculate the precise distance. The candidates with the minimum acoustic distance $D(k)$ are indicated as the search results. Since the $R(k)$ is calculated as an approximate value of DP matching distance $D(k)$, after L_{I_c} is sorted by $R(k)$, the correct lyric with the minimum $D(k)$ rises into the forward ranks in most cases. Thus, instead of the exhaustive DP matching over the entire set of pre-selected lyrics L_{I_c}, a DP-based search with an adaptive termination criterion is proposed. The termination is adaptive to a cut off function F. The second pass search is designed as shown in the flowchart in Fig. 4.

Lyrics L_{I_c} are first sorted by $R(k)$ and then divided into Z groups, thus each group has I_c/Z lyrics. A DP matching calculation is executed in one group after another, while the cut-off function F does not fulfill a terminating condition. Once the value of F reaches a threshold F_{th}, the DP matching process is aborted at that group. Within the lyrics of the calculated groups, the lyrics are ranked in the order of the $D(k)$, and then the lyrics with lower distance values are provided as search results.

6. Experiments

Two segments of experiments were carried out in this paper. First, the improvement of search accuracy by applying acoustic distance was evaluated. Second, the proposed method applying the acoustic distance and the two-pass search algorithm was compared with three conventional methods to evaluate its performance on both search accuracy and processing time.

The results of the experiments were all obtained using a personal computer, with the specifications of Intel Core2Duo CPU 3.0GHz and 4G RAM.

6.1 Verification of Improvements in Search Accuracy by Applying the Acoustic Distance

Two exhaustive DP-based search methods using different distances were compared to evaluate the advantage of the acoustic distance. One method is Exhaustive DP applying Edit Distance (EDPED) of phoneme strings, the other method is Exhaustive DP applying Acoustic Distance (EDPAD).

The test set consisted of 220 incorrect queries that were mistaken via acoustic confusion, the same as the queries used in Sect. 3.2. Also, a database of 10,000 lyric texts was collected. It contained both Japanese and English lyrics. The lyrics corresponding to the queries were included in the database.

As shown in Table 3, T-best (T = 1, 20) represents the top T candidates of the ranked lyrics. The hit rate of T-best is defined as the rate of the total number of hits within top T candidates to the total number of search accesses (this is calculated as the search accuracy). EDPAD improves the hit rates by 2.8% and 4.4% respectively when the value of T in T-best is 1 and 20. A t-test was also conducted. When T in T-best is 20, the p-values are very low, which indicates that the proposed acoustic distance method achieved statistically significantly better performances than edit distance. Futhermore, an analysis of the queries that failed to identify the target lyric text using EDPAD method reveals that most of them are smaller than 6 syllables, indicating that the distance between the query and lyric texts was too close to

Table 3 Search accuracies and p-values of EDPED and EDPAD.

Search methods		EDPED	EDPAD
1-best	hit rate(%)	49.5	50.9
	p-value	0.203	
20-best	hit rate(%)	70.5	73.6
	p-value	0.017	

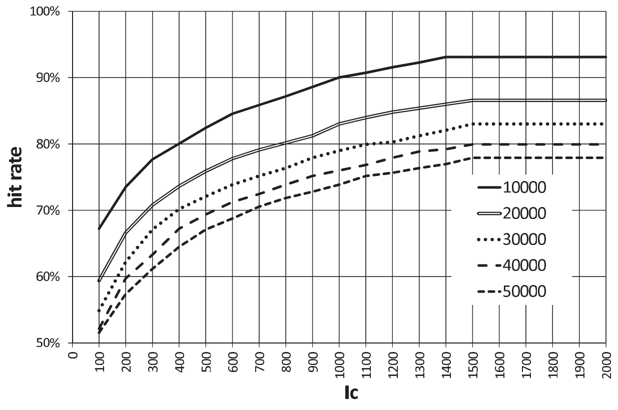

Fig. 5 Rlationship between hit rates and I_c for various sizes of lyric database.

make the target lyric distinguishable.

6.2 Evaluation of Search Accuracy and Search Time

The total performance of the proposed method applying the acoustic distance and the two-pass search algorithm, which are described in Sect. 4 and 5, is evaluated by lyric search experiments in this section.

Before the comparison experiment, in order to optimize the parameters of the proposed method, preliminary experiments were carried out.

6.2.1 Preliminary Experiments to Determine Parameters for the First and the Second Passes

As described in Sect. 5, the following parameters in the proposed method need to be optimized:

- I_c: the number of candidates in the first pass
- F: the cut-off function in the second pass

To find corpus-independent parameters, 842 misheard lyric queries in English were collected from the website "kissthisguy". Also, a database of 50,000 lyric texts was collected. The lyrics corresponding to the queries were all included in the database. Note that, the queries and lyric texts are entirely different from those used in Sect. 6.1.

First, an experiment was carried out to decide I_c. The first pass search using the index described in Sect. 5.2 was executed to investigate the relationship between search accuracy and I_c to choose the best value for I_c. The results are shown in Fig. 5. The horizontal axis shows the values of each tested I_c from 100 to 2,000, and the vertical axis is the hit rate within I_c candidates. Each line represents a different number of lyrics in the search space. The hit rates are almost saturated when I_c is larger than 1500, in spite of the variation in the search space. Therefore, I_c is set at 1500 in this paper.

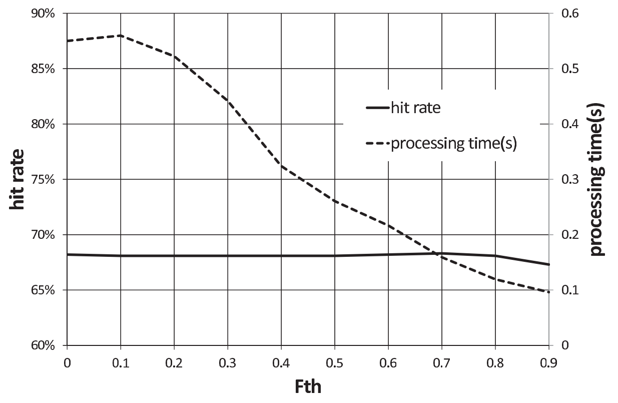

Fig. 6 Search accuracy and processing time with respect to F_{th} in the case of 1-best.

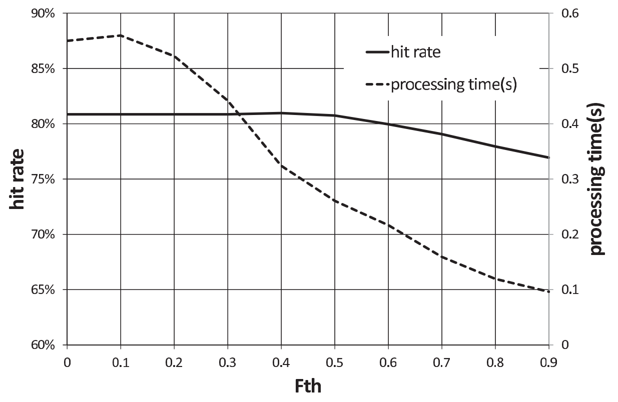

Fig. 7 Search accuracy and processing time with respect to F_{th} in the case of 20-best.

Second, an investigation was undertaken to decide F. In most of the 842 queries, it was found that, by sorting the lyrics according to the approximate distance $R(k)$ and dividing them into groups, the target lyric has a significantly lower DP distance $D(k)$ than other lyrics in the same group. Based on the investigation above, F is defined by Eq. (12), where D_{min} is the minimum value and D_{mean} is the mean value of the group. The experimental results that reveal the relationship between the processing time and search accuracy with respect to F_{th} are shown in Fig. 6 and Fig. 7, where the horizontal axis represents F_{th}, the right vertical axis represents processing time, and the left vertical axis represents the hit rate. Figure 6 shows the results for the 1-best case, while Fig. 7 shows the results for the 20-best case. Both figures show that the value of F_{th} between 0.4 ∼ 0.6 is the optimal threshold to reduce processing time without deteriorating search accuracy.

$$F = \frac{D_{min}}{D_{mean}} \tag{12}$$

6.2.2 Evaluation of the Overall Performance

To evaluate the overall performance of the proposed method, both hit rate and processing time were compared with three conventional DP-based methods. All methods applied the proposed acoustic distance. The details are described below.

- "Two-pass DP search with Adaptive Termination (TDPAT)" is the proposed method described in Sect. 4 and 5. F_{th} is tuned from 0 to 1. Considering the balance of index size and search accuracy, here N of the syllable N-gram index is set to 3. A total of 50,000 entries of syllable 3-grams, which cover 92% of all syllable 3-grams in the collected lyric corpus, are prepared in the index. As all the syllable 3-grams which exist in the queries are prepared, no search errors come from out-of-vocabulary syllable 3-grams in the experiment. The acoustic distance is normalized by the length of the corresponding DP path. The group size for the second pass is set at 100 lyrics, as Z is 15. It is optimized by a preminary experiment.
- "EDPAD" is an exhaustive DP-based search over the entire search space of lyrics, and this method is mentioned in Sect. 6.1.
- "High-speed DP search with Suffix Array (HDPSA)"is based on the method in [8]. In the experiment, since the input query and the database are both text, the texts were converted into syllable strings (instead of the phonemes originally used); and divided into syllable N-grams. Also, a suffix array recorded the boundary information of the lyrics in order to avoid matching queries across two lyrics. Here N is set to 3 because this value resulted in better performance than when $N = 2$ or $N = 4$ in a preminary experiment. The total threshold was tuned from 0 to 1.3 to find the optimal value balancing search accuracy and processing time.
- "Two-pass DP search with Distance-based Termination (TDPDT)" is a method that has almost the same processes as the proposed method, with the exception that the DP is terminated when the acoustic distance $D(k)$ exceeds a predetermined threshold value, that is tuned from 0 to 1.

The test set of 220 incorrect queries and 10000 lyric texts were the same with those used in Sect. 6.1.

First, to evaluate the robustness of TDPAT, a comparison with EDPAD and TDPAT is represented in Fig. 8. Here, F_{th} for TDPAT is set at 0.4, as optimized in Sect. 6.2.1. TDPAT maintains almost the same hit rate as the T of T-best is varied from 1 to 40. As T is over 40, there is also only less than 1.7% deterioration of search accuracy. As the processing time of TDPAT is 0.23 seconds per query, it is reduced by 89.3% compared with EDPAD. This improvement is due to the well-designed two-pass search algorithm that avoids losses occurring in the pre-selection and the adaptive termination processes.

The search accuracy and time complexity of three high-speed DP methods TDPAT, TDPDT and HDPSA are shown in Fig. 9 and Fig. 10, where the horizontal axis represents processing time and the vertical axis represents hit rate. Each point in these figures indicates the processing time cost and the hit rate achieved when a particular threshold is set. Figure 9 and Fig. 10 show the results in the cases of 1-best and 20-best, respectively.

As shown in both figures, the performance of TDPAT is superior to that of HDPSA in terms of both processing time

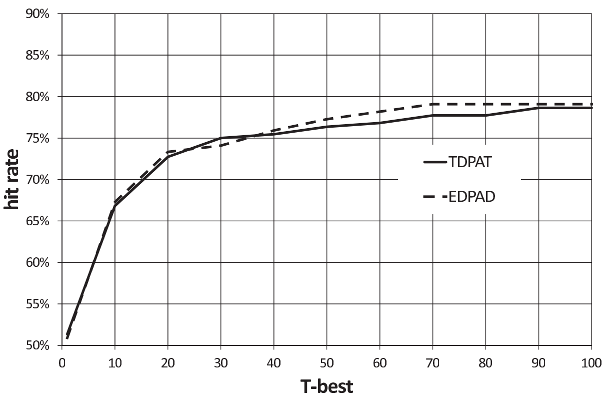

Fig. 8 Search accuracy of TDPAT and EDPAD.

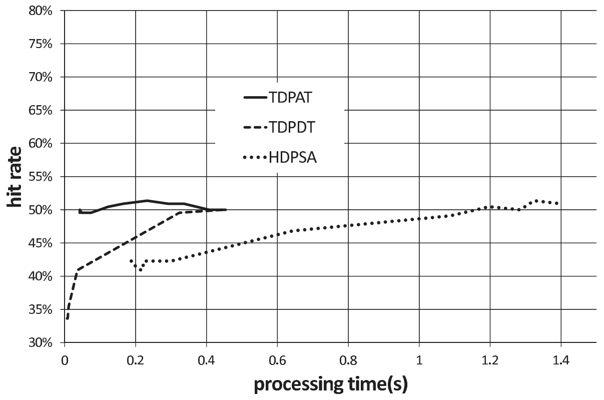

Fig. 9 Average processing times and search accuracy of three search methods in the case of 1-best.

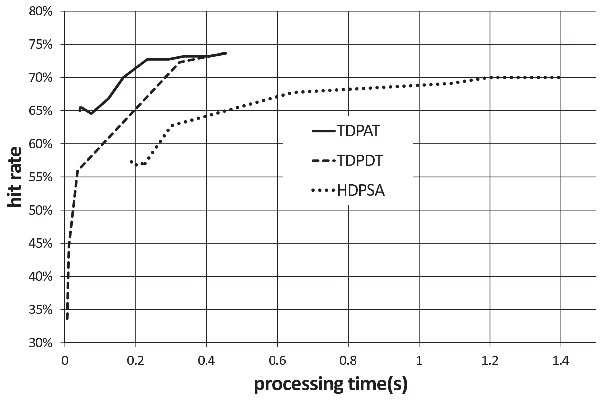

Fig. 10 Average processing times and search accuracy of three search methods in the case of 20-best.

and search accuracy. In the case of 1-best, to achieve the same hit rate of 50.0%, TDPAT reduces processing time by a maximum of 96.5% compared with HDPSA. In the case of 20-best, to achieve the same hit rate of 70.0%, TDPAT reduces processing time by a maximum of 86.2%. These results indicate that the proposed search algorithm is more efficient than the conventional algorithms those determine the pruning threshold according to the length of the queries.

Also, TDPAT obtains higher search accuracy than TD-PDT at the same processing times, especially for short processing times. It proves that the hypothesis of the definition for F is correct and is effective in the search process.

7. Conclusions

This paper proposed a robust and fast lyric search method based on acoustic distance and a two-pass search strategy using an index-based approximate preselection for the first pass and a DP-based string matching in the second pass. For the incorrect queries that are misheard or mismemorized, the experiments proved that applying acoustic distance improved search accuracy by 4.4% over edit distance. Though the search accuracy is expected to be more improved if the acoustic distances are calculated from the singing voice data, the proposed method offered a realistic and efficient solution with an easily-obtainable database of more general ordinary speech. Furthermore, the proposed method achieved real-time operation by reducing processing time by more than 86.2% with a slight loss in search accuracy compared with a complete search by DP matching with all lyrics. It is proved to be the most practical solution for acoustic confusion queries, considering the trade-off between high search accuracy and low computation complexity.

References

[1] Downie and Cunningham, "Toward a theory of music information retrieval queries: System design implications," Proc. ISMIR 2009, pp.299–300, 2002.

[2] J. Zobel and P. Dart, "Phonetic string matching: Lessons from information retrieval," Proc. 19th International Conference on Research and Development in Information Retrieval, pp.166–172, 1996.

[3] V.T. Turunen and M. Kurimo, "Indexing confusion networks for morph-based spoken document retrieval," Proc. 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.631–638, 2007.

[4] T. Hori, I.L. Hetherington, T.J. Hazen, and J.R. Glass, "Open-Vocabulary Spoken Utterance Retrieval Using Confusion Networks," Proc. 2007 International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.73–76, 2007.

[5] N. Ring and A. Uitenbogerd, "Finding 'Lucy in Disguise': The Misheard Lyric Matching Problem," Proc. AIRS 2009, pp.157–167, 2009.

[6] H. Hirjee and D.G. Brown, "Solving misheard lyric search queries using a probabilistic model of speech sounds," Proc. ISMIR 2010, pp.137–148, 2010.

[7] http://www.kissthisguy.com/

[8] K. Katsurada, S. Teshima, and T. Nitta, "Fast keyword detection using suffix array," Proc. INTERSPEECH, pp.2147–2150 2009.

[9] T. Yamasita and Y. Matsumoto, "Full text approximate string search using suffix arrays," IPSJ SIG Technical Reports, 1997-NL-121, pp.23–30, 1997. (In Japanese)

[10] http://okwave.jp/

[11] http://oshiete.goo.ne.jp/

[12] C. Fox, "A stop list for general text," ACM SIGIR Forum, vol.24, no.1–2, pp.19–21, Fall 1989/Winter 1990.

[13] N. Kummer, C. Womser-Hacker, and N. Kando, "MIMOR@NTCIR 5: A fusion-based approach to Japanese information retrieval," Proc. NTCIR-5 Workshop Meeting, Tokyo, Japan, 2005.

[14] D. Oshyvanyk, Y.-G. Gueheneuc, A. Marcus, G. Antoniol, and V. Rajlich, "Combining probabilistic ranking and latent semantic indexing for feature identification," 14th IEEE International Conference on Program Comprehension, pp.137–148, 2006.

[15] M. Yamada, T. Kato, M. Naito, and H. Kawai, "Improvement of rejection performance of keyword spotting using anti-keywords derived from large vocabulary considering acoustical similarity to keywords," Proc. INTERSPEECH, pp.1445–1448, 2005.

[16] http://mecab.sourceforge.net

Xin Xu received his B.E. degree in electrical engineering from Zhejiang University, China, in 2004 and B.E. degree in information science and technology from Hokkaido University in 2006. He joined KDDI Co. Ltd. in 2007. He is currently with KDDI R&D Laboratories, Inc. His research interests are music information retrieval and spoken dialogue systems. He is a member of IEICE.

Tsuneo Kato received his B.E., M.E. and Ph.D. from The University of Tokyo in 1994, 1996, and 2011, respectively. He joined Kokusai DenshinDenwa Co. Ltd. in 1996. He is currently with KDDI R&D Laboratories, Inc. He has been engaged in research and development of automatic speech recognition, spoken dialogue systems, and interactive user interface. He received IPSJ Kiyasu Special Industrial Achievement Award in 2011. He is a member of ASJ, IPSJ, IEEE, and IEICE.