# SONGLE: A WEB SERVICE FOR ACTIVE MUSIC LISTENING IMPROVED BY USER CONTRIBUTIONS

**Masataka Goto, Kazuyoshi Yoshii, Hiromasa Fujihara, Matthias Mauch, and Tomoyasu Nakano**
National Institute of Advanced Industrial Science and Technology (AIST), Japan

## ABSTRACT

This paper describes a public web service for active music listening, *Songle*, that enriches music listening experiences by using music-understanding technologies based on signal processing. Although various research-level interfaces and technologies have been developed, it has not been easy to get people to use them in everyday life. Songle serves as a showcase to demonstrate how people can benefit from music-understanding technologies by enabling people to experience active music listening interfaces on the web. Songle facilitates deeper understanding of music by visualizing music scene descriptions estimated automatically, such as music structure, hierarchical beat structure, melody line, and chords. When using music-understanding technologies, however, estimation errors are inevitable. Songle therefore features an efficient error correction interface that encourages people to contribute by correcting those errors to improve the web service. We also propose a mechanism of *collaborative training for music-understanding technologies*, in which corrected errors will be used to improve the music-understanding performance through machine learning techniques. We hope Songle will serve as a research platform where other researchers can exhibit results of their music-understanding technologies to jointly promote the popularization of the field of music information research.

## 1. INTRODUCTION

The goal of this research is to enrich music listening experiences by using music-understanding technologies based on signal processing. Toward this goal, we have already developed various *active music listening interfaces* [1], where active music listening is a way of listening to music through active interactions. In this research, the word *active* is not meant to convey that the listeners create new music, but that they take control of their own listening experience. For example, the active music listening interface *SmartMusicKIOSK* [2] has a chorus-search function that enables a user to access directly his or her favorite part of a song (and to skip others) while viewing a visual representation of its music structure, which facilitates deeper understanding. However, up to now the general public has not had the chance of using such research-level interfaces and technologies in daily life.

We therefore developed a web service called *Songle* that allows anonymous web users to enjoy music by using active music listening interfaces on a web browser. Songle uses automatic music-understanding technologies to estimate music scene descriptions (musical elements) [3, 4] of musical pieces (audio files) available on the web. A user of Songle can enjoy playing back a musical piece while seeing the visualization of the estimated descriptions. In our current implementation, four major types of descriptions are automatically estimated and visualized for content-based music browsing: music structure (chorus sections and repeated sections), hierarchical beat structure (musical beats and bar lines), melody line (fundamental frequency (F0) of the vocal melody), and chords (root note and chord type). In particular, Songle implements all functions of the SmartMusicKIOSK interface, and a user can jump and listen to the chorus with just a push of the next-chorus button. Songle thus makes it easier for a user to find desired parts of a piece.

Given the variety of musical pieces on the web, however, it is difficult to estimate music scene descriptions with high accuracy. Because of the diversity of music genres, complexity of sound mixtures, and recording conditions, automatic music-understanding technologies cannot avoid making some errors, even though the technologies are constantly improving. As a result, users of such a web service might be disappointed by its performance.

To overcome this difficulty, Songle enables anonymous users to contribute by correcting music-understanding errors. Each user can see the music-understanding visualizations on a web browser, with a moving cursor indicating the audio playback position. If a user finds an error while listening, the user can easily correct the error by selecting from a list of candidates, or by providing an alternative description on Songle's efficient error correction interface. The resulting corrections are then shared and used to immediately improve the user experience with the corrected piece. We also plan to use such corrections to gradually improve music-understanding technologies through adaptive machine learning techniques, so that descriptions of other musical pieces can be estimated more accurately. This approach can be described as *collaborative training for music-understanding technologies* on the web.

Development for Songle started in June 2009, and the Songle website *http://songle.jp* will be open to the public

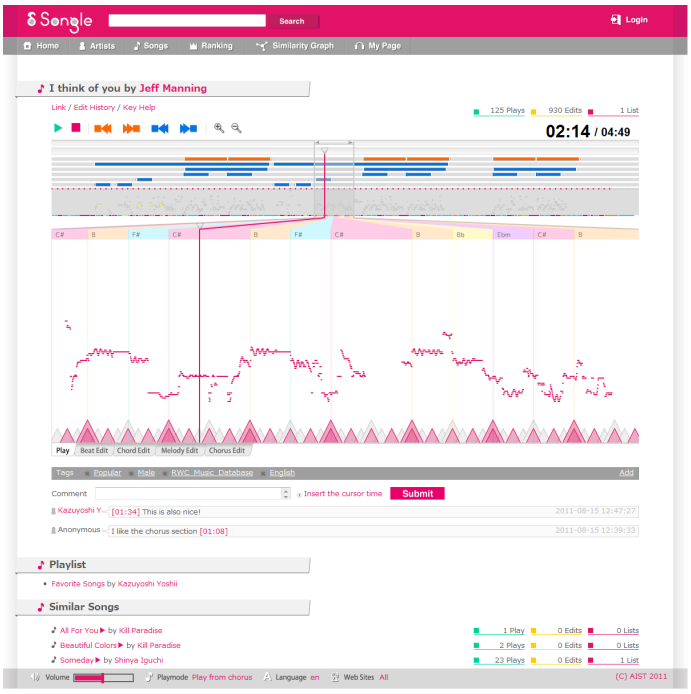



**Figure 1**. Songle screen snapshot of the main interface for music playback with the visualization of music scene descriptions estimated automatically.

before the ISMIR 2011 conference. In addition to the contribution of enriching music listening experiences, Songle will serve as a showcase in which everybody can experience music-understanding technologies and understand their nature: for example, what kinds of music or sound mixture are difficult for the technologies to handle. Furthermore, we hope to extend Songle so that it can serve as a research platform able to support various music-understanding technologies developed by different researchers.

## 2. OVERVIEW OF SONGLE

Songle is a social annotation web service where users can retrieve, browse, and annotate musical pieces on the web. Figure 1 shows the web page of the Songle interface after a musical piece is selected. During the initial stage of the Songle launch we are focusing on popular songs with vocals. Songs recently released on music web services such as Magnatune (http://magnatune.com/) and PIAPRO[1] are added (registered) to Songle. A user can also register any song available on the web by providing the URL of its MP3 file, the URL of a web page including multiple MP3 URLs, or the URL of a music podcast (an RSS syndication feed including multiple MP3 URLs).

Everybody can enjoy active music listening and correct errors as an anonymous user without logging in, but a user has to log in with OpenID to register a new song. In addition, a logged-in user can generate a playlist. Such a playlist has a title (theme) and the user can choose whether to share it with other users. A logged-in user can also enter and record a preference (like or dislike) for each song to get better song recommendations in the future.

Songle supports three main functions: retrieving, browsing, and annotating songs. The retrieval and browsing functions facilitate deeper understanding of music, and the annotation (error correction) function allows users to contribute to improve music scene descriptions. These improved descriptions can then lead to a better user experience of retrieving and browsing songs.

### 2.1 Retrieval Function

This is a function that enables a user to retrieve a song through a text search of the song title or artist name, or through selection from a list of artists or a list of songs whose descriptions were recently estimated or corrected. This function also shows various kinds of ranking, such as artist ranking, song ranking, and user ranking (by the number of corrected errors).

Following the idea of the active music listening interface *VocalFinder* [5], which finds songs with similar vocal timbres, a similarity graph of songs is also visualized so that a user can retrieve a song according to vocal timbre similarity. The graph is a radially-connected network in which nodes (songs) of similar vocal timbre are connected to the center node (a recommended or user-specified song). By traversing a graph while listening to nodes, a user can find a song having the favorite vocal timbre.

A user can play back a song in any list of songs (e.g., playlist, retrieved list, and ranking) for trial listening to judge whether it is of interest. By selecting one of the songs, the user switches over to the next browsing function. While using the browsing function, songs in the playlist are automatically switched one after another if a user selects and listens to a playlist.

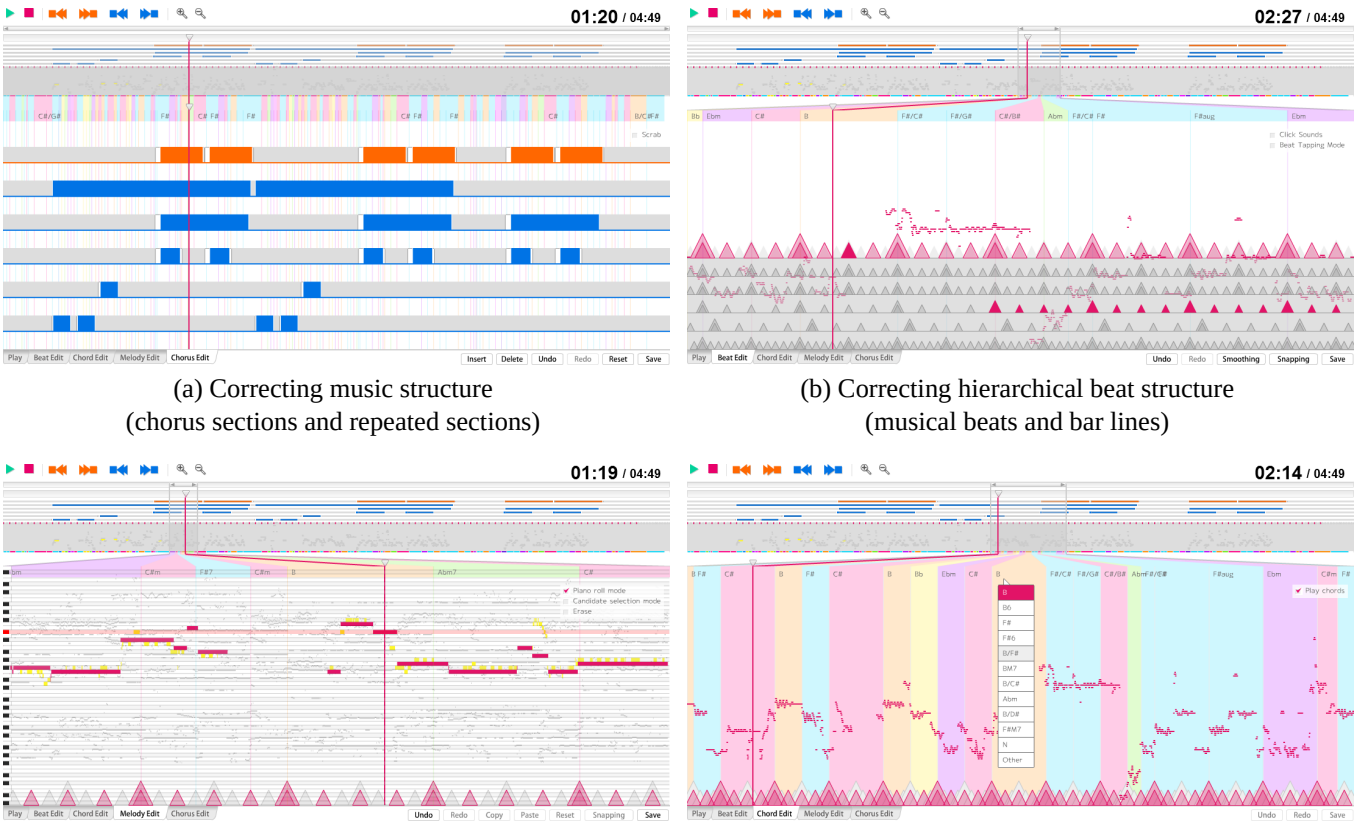
### 2.2 Within-song Browsing Function

This is a function that provides a content-based playback-control interface for within-song browsing as shown in the upper half of Figure 1. The upper window is the global view showing the entire song and the lower window is the local view magnifying the selected region.

With this function, a user can view the following four types of music scene descriptions estimated automatically:

1. *Music structure (chorus sections and repeated sections)*
   In the global view, the *music map* of the SmartMusicKIOSK interface [2] is shown below the playback controls including the buttons, time display, and playback slider. The music map is a graphical representation of the entire song structure consisting of chorus sections (the top row) and repeated sections (the five lower rows). On each row, colored sections indicate similar (repeated) sections. This map helps a user decide where to jump. Clicking directly

[1] PIAPRO (http://piapro.jp/) is a web service to which musicians can upload their own songs created using singing synthesizers.




(a) Correcting music structure (chorus sections and repeated sections)

(b) Correcting hierarchical beat structure (musical beats and bar lines)

(c) Correcting melody line (F0 of the vocal melody)

(d) Correcting chords (root note and chord type)

**Figure 2**. Songle screen snapshots of the annotation function for correcting music scene descriptions.

on a colored section plays that section. There are also buttons for jumping to the next or previous chorus sections, and the next or previous repeated sections.

2. *Hierarchical beat structure (musical beats and bar lines)*
   At the bottom of the local view, musical beats corresponding to quarter notes are visualized by using small triangles. The top of each triangle indicates its temporal position. Bar lines are marked by larger triangles.
3. *Melody line (F0 of the vocal melody)*
   The piano roll representation of the melody line is shown above the beat structure in the local view. It is also shown in the lower half of the global view. For simplicity, the fundamental frequency (F0) can be visualized after being quantized to the closest semitone.
4. *Chords (root note and chord type)*
   Chord names are written in the text at the top of the local view. Twelve different colors are used to represent twelve different root notes so that a user can notice the repetition of chord progressions.

In the lower half of Figure 1, a user can add and share social tags and time-synchronous comments for *Crowd Music Listening* [6]. Clicking on a time-synchronous comment starts playback from that position.

### 2.3 Annotation (Error Correction) Function

This function allows users to add annotations to correct any estimation errors they may come across while listening to music. Here, annotation means describing the contents of a song, either by modifying the estimated descriptions or by selecting the correct candidate if available. For this purpose, we provide an efficient error correction interface (editor) as shown in Figure 2.

Editors for four types of music scene descriptions can be switched between in the local view.

1. *Music structure* (Figure 2(a))
   The beginning and end points of every chorus or repeated section can be adjusted. It is also possible to add, move, or delete each section. Repeated sections can serve as candidates for the chorus sections, but since it is not obvious how to correct repeated sections, we plan to let a user type in a section label (e.g., verse A, verse B, etc.) on each row. This correction function improves the SmartMusicKIOSK experience.
2. *Hierarchical beat structure* (Figure 2(b))
   Several alternative candidates for the beat structure can be selected at the bottom of the local view. If none of the candidates are useful, a user can enter the beat position by tapping a key during music playback. Each beat position or bar line can also be changed directly. For fine adjustment it is possible to play back the audio with click tones at beats.
3. *Melody line* (Figure 2(c))
   Songle allows note-level correction on the piano roll rep-

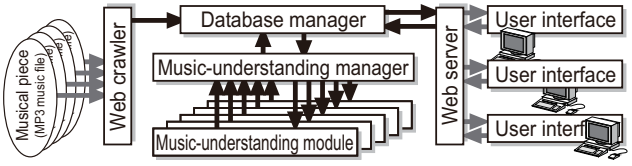



**Figure 3**. Implementation overview of Songle.

resentation of the melody line. Since the melody line is internally represented as the temporal trajectory of F0, more precise correction is also possible by choosing from F0 candidates. A user can listen to the melody line only or the melody-cancelled background playback. More accurate melody annotations will lead to better similarity graphs of songs.

4. *Chords* (Figure 2(d))
   Chord names can be corrected by choosing from candidates or by explicit typing of chord names. Each chord boundary can also be adjusted. Chords can be played back along with the original song to make it easier to check the correctness.

Note that users can simply enjoy active music listening without correcting errors. We understand that it is too difficult for some users to correct the above descriptions (especially, chords). Designing an interface that makes it easier for them to correct will be another future challenge. Moreover, users are not expected to correct all errors, only some according to each user's interests.

When the music-understanding results are corrected by users, the original values are visualized as trails with different colors (white, gray, or yellow marks in Figure 2) that can be distinguished by anybody. These trails are important to prevent overestimation of the automatic music-understanding performance after the user corrections. Moreover, all the correction histories are recorded, and descriptions before and after corrections can be compared.

## 3. IMPLEMENTATION OF SONGLE

The implementation overview of Songle is shown in Figure 3. The *web crawler* collects musical pieces (MP3 files) on the basis of their URLs and RSS feeds, which can be added by users, and stores the pieces in the database. Several *music-understanding modules*, each corresponding to a particular type of music scene description, then process each musical piece. For example, the beat structure and the music structure are estimated by two different modules. When a request from an idle music-understanding module is received by the *music-understanding manager*, the next available musical piece lacking an estimation result for the corresponding description is handed over. After the music-understanding module finishes processing its piece, the estimation result is passed to the database manager via the music-understanding manager. The *database manager* controls the processing state of the musical pieces and stores their estimation results in a database together with the corrections when provided by users. Finally, the *web server* works as a website that provides the Songle *user interface*, which directly plays back the MP3 file from the original URL.

The web server of Songle was implemented by using a web application framework *Ruby on Rails*, a programming language *Ruby*, a web server *Passenger* and *Apache*, and a database *MySQL*. The client user interface was implemented by using a scripting language *ActionScript 3*, an ActionScript 3 compiler *Adobe Flex Compiler*, and a scripting language *JavaScript*.

Music scene descriptions are estimated as follows.

1. *Music structure*
   Chorus sections and repeated sections are estimated by using the chorus-section detection method *RefraiD* [2] which focuses on popular music. By analyzing relationships between various repeated sections, the RefraiD method can detect all the chorus sections in a song and estimate both ends of each section. It can also detect modulated chorus sections.
2. *Hierarchical beat structure*
   The beats are estimated using a hidden Markov model (HMM) with 43 tempo states, each having 18 to 60 sub-states corresponding to the beat phase of different tempi. In each tempo a beat is modeled as a left-to-right HMM in which only some states have non-deterministic transition probabilities to allow for tempo fluctuations or tempo changes. The emission probability of a sub-state is calculated via the cosine similarity between a comb filter and a simple onset detection function. Five different comb-filter shapes are used to output five different beat-tracking results, including those that are likely true candidates if the default algorithm tracks the back-beat or beats at twice the true tempo. This strategy maximizes the likelihood of offering the beat-tracking result desired by the user.
   The bar lines are estimated using harmonic cues. First, we extract tatum-synchronous bass and treble chromagrams using NNLS Chroma [7]. We build a simple chord detection model and calculate posterior probabilities of chord changes. Using a sliding window, we compute the cosine similarity between the chord change probabilities and several different bar patterns that cover the 3/4, 4/4 and 6/8 meters and all possible bar phases. We normalize the cosine similarities at each frame and use them as emissions in another HMM similar to the beat-tracking model.
3. *Melody line*
   The fundamental frequency (F0) of the vocal part is estimated by using the F0 estimation method for the vocal melody [8], which is implemented by extending the predominant-F0 estimation method *PreFEst* [4]. This method focuses only on the vocal melody by evaluating a GMM-based vocal probability for each F0 candidate estimated by PreFEst. Moreover, vocal activity detection was implemented by using a method described in [9].
4. *Chords*
   We transcribe chords using 14 chord types: major, major 6th, major 7th, dominant 7th, minor, minor 7th, half-

diminished, diminished, augmented, and five variants of major chords with different bass notes: /2, /3, /5, /b7, and /7. The resulting 14 types $\times$ 12 root notes = 168 chords and one 'no chord' label are estimated using an HMM approach on the same tatum-synchronous chromagram used for the bar-line estimation. Chord changes are allowed to happen only on beats.

We also include knowledge from the bar-line estimation and a key estimate. Instead of building one large model in which chords and keys are modeled simultaneously (e.g. [10]), we chose a less memory-intensive, progressive approach. First, we model the key in a simple separate HMM with three different key scales: major, natural minor, and harmonic minor. Every key state has observation probabilities for all different chords, based on an expert function [10]. The posterior probability obtained from the HMM is then used to weight the chord probabilities for the chord HMM. During Viterbi decoding we use the bar-line estimates for dynamic transition probability weighting in order to encourage chord changes at bar lines.

To show the similarity graph of songs, the vocal timbre similarity between songs is calculated by using the method used in VocalFinder [5]. The current recommendation of songs simply uses the same similarity, which shall be improved in the future.

## 4. DISCUSSION

While research dealing with a public web service for active music listening and social annotation has not been pursued in the past, there have been various approaches related to music annotation. In the following, we introduce related work and then discuss how Songle could contribute to society and academic research.

### 4.1 Related Research

Several approaches have been proposed to collect a large amount of ground-truth annotations. Such annotations are useful for improving the accuracy of music information retrieval (MIR) systems using machine learning techniques, and for evaluating MIR systems [11]. For example, Lee [12] used a web service called Amazon Mechanical Turk (MTurk) to ask people to make similarity judgments. Mandel *et al.* [13] also used MTurk for tag collection. These approaches showed that the quality of the collected annotations was sufficiently high. However, the human effort necessary increases in proportion to the required number of annotations.

To solve this problem, it is interesting to let non-experts contribute to making ground-truth annotations. One promising approach is based on games. For example, Turnbull *et al.* [14, 15] proposed an annotation game, where players are asked to choose the most and least suitable descriptions for a given musical piece. Mandel and Ellis [16] proposed another annotation game, where players can get points by providing useful descriptions that were not provided by other players. Law *et al.* [17] proposed another annotation game based on the ESP Game [18] in which players are not asked to describe a sound, but told to guess what their randomly paired partners are thinking. Songle provides a stronger motivation for users to contribute than these related approaches because the user is aware of improving the service for other users.

To annotate musical pieces, various useful editors have been developed, such as Sonic Visualiser [19], Audacity extension [20], CLAM [21], and MUCOSA [22]. The Echo Nest API (http://developer.echonest.com/) is also useful for access to various annotations. Songle is the first system that allows anonymous users to collaborate to edit various music scene descriptions (chorus, beats, melody, and chords) directly on the web without stand-alone applications.

### 4.2 Contributions of Songle

Songle makes a social contribution by providing the world's first public web service for enjoying active music listening interfaces with music-understanding technologies. It also promotes the popularization and use of music-understanding technologies by raising user awareness. Users can grasp the nature of music-understanding technologies just by seeing results of the technologies applied to songs available on the web. When there are many errors, we run the risk of attracting criticism, but we believe that sharing these results with users will promote further popularization of this research field.

The academic contribution of this study is to propose a new research approach to music understanding based on signal processing; this approach aims at improving both the music-understanding performance and the usage rate while benefiting from the cooperation of anonymous end users. This approach is designed to set into motion a *positive spiral* where (1) we enable users to experience a service based on music understanding to let them better understand its performance, (2) users contribute to improved performance, and (3) the improved performance leads to a better user experience, which encourages further use of the service at step (1) of this spiral. This is a *social correction* framework, where users can improve the performance by sharing their correction results over a web service. The game-based approach of Human Computation or GWAPs (games with a purpose) [23] like the ESP Game [18] often lacks step (3) and depends on the feeling of fun. In this framework, users gain a real sense of contributing for their own benefit and that of others and can be further motivated to contribute by seeing corrections made by other users. In this way, we can use the *wisdom of crowds* or *crowdsourcing* to achieve a better user experience.

Another important technical contribution of this study is to investigate how far the performance of music-understanding technologies can be improved by getting errors corrected through the cooperative efforts of users. Although we have not yet implemented a machine-learning mechanism to improve the performance on the basis of user corrections, we could implement such a mechanism once we

have collected enough corrections. This study will then provide a framework for *amplifying* user contributions in music research. In a typical *Web 2.0* service like *Wikipedia*, improvements are limited to an item directly contributed (edited) by users. In Songle, improvements will automatically spread to other songs because of the improvement of the music-understanding technologies through machine learning techniques. This will be a novel technology of amplifying user contributions, which could be beyond Web 2.0 and Human Computation [23]. We hope that this study will show the importance and potential of incorporating and amplifying user contributions in music research, as have been demonstrated by PodCastle in speech research [24, 25].

We think we can trust users with respect to the quality of correction according to our experiences from PodCastle [25]. Even if some users deliberately make inappropriate corrections (the vandalism problem), we will be able to develop countermeasures to acoustically evaluate the reliability of corrections. For example, we could validate whether the corrected descriptions can be supported by acoustic phenomena. This will be another interesting topic of research.

### 4.3 Songle as a Research Platform

In the future, we hope to extend Songle to serve as a research platform where other researchers can also exhibit results of their own music-understanding technologies. When each technology is implemented as a *music-understanding module* in Figure 3, which can be executed anywhere in the world even in our current implementation, it is not necessary to share its source and binary codes. Each in-house module, even inside an Internet firewall, can just connect to the *music-understanding manager* to receive an audio file and send back music-understanding results via HTTP. The results should always be shown with clear acknowledgments/credits so that users can distinguish the sources. We are also interested in adding other types of music scene descriptions, which are not yet supported.

Once this happens, it will be interesting to compare/visualize differences of modules on each music scene description. Results from different researchers can also be used as candidates for the correction and even as votes to let different results converge.

## 5. CONCLUSION

We have described Songle, an active music listening service that is continually improved by user contributions. In our current implementation, four types of music scene descriptions are estimated and exposed through web-based interactive user interfaces. Since automatic music-understanding technologies are not perfect, Songle allows users to make error corrections, which are shared with other users, thus creating a positive spiral and an incentive to keep correcting. For the MIR community this platform will act both as a test-bed or showcase for new technologies, and as a way of collecting valuable annotations.

**ACKNOWLEDGMENTS:** We thank Utah Kawasaki for the web service implementation and Minoru Sakurai for the web design of Songle. This work was supported in part by CrestMuse, CREST, JST.

## 6. REFERENCES

[1] M. Goto, "Active music listening interfaces based on signal processing," in *Proc. of ICASSP 2007*, 2007.
[2] M. Goto, "A chorus-section detection method for musical audio signals and its application to a music listening station," *IEEE Trans. on ASLP*, vol. 14, no. 5, pp. 1783–1794, 2006.
[3] M. Goto, "Music scene description project: Toward audio-based real-time music understanding," in *Proc. of ISMIR 2003*, pp. 231–232, 2003.
[4] M. Goto, "A real-time music scene description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.
[5] H. Fujihara *et al.*, "A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval," *IEEE Trans. on ASLP*, vol. 18, no. 3, pp. 638–648, 2010.
[6] M. Goto, "Music listening in the future: Augmented Music-Understanding Interfaces and Crowd Music Listening," in *Proc. of the AES 42nd International Conf. on Semantic Audio*, pp. 21–30, 2011.
[7] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proc. of ISMIR 2010*, pp. 135–140, 2010.
[8] H. Fujihara *et al.*, "F0 estimation method for singing voice in polyphonic audio signal based on statistical vocal model and Viterbi search," in *Proc. of ICASSP 2006*, pp. V–253–256, 2006.
[9] H. Fujihara *et al.*, "Automatic synchronization between lyrics and music CD recordings based on Viterbi alignment of segregated vocal signals," in *Proc. of ISM 2006*, pp. 257–264, 2006.
[10] M. Mauch and S. Dixon, "Simultaneous estimation of chords and musical context from audio," *IEEE Trans. on ASLP*, vol. 18, no. 6, pp. 1280–1289, 2010.
[11] J. S. Downie, "The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research," *Acoustical Science and Technology*, vol. 29, pp. 247–255, 2008.
[12] J. H. Lee, "Crowdsourcing music similarity judgments using Mechanical Turk," in *Proc. of ISMIR 2010*, pp. 183–188, 2010.
[13] M. I. Mandel, D. Eck and Y. Bengio, "Learning tags that vary within a song," in *Proc. of ISMIR 2010*, pp. 399–404, 2010.
[14] D. Turnbull *et al.*, "A game-based approach for collecting semantic annotations of music," in *Proc. of ISMIR 2007*, pp. 535–538, 2007.
[15] D. Turnbull, L. Barrington and G. Lanckriet, "Five approaches to collecting tags for music," in *Proc. of ISMIR 2008*, pp. 225–230, 2008.
[16] M. I. Mandel and D. P. W. Ellis, "A Web-based game for collecting music metadata," in *Proc. of ISMIR 2007*, pp. 365–366, 2007.
[17] E. L. M. Law *et al.*, "TagATune: A game for music and sound annotation," in *Proc. of ISMIR 2007*, pp. 361–364, 2007.
[18] L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proc. of CHI 2004*, pp. 319–326, 2004.
[19] C. Cannam *et al.*, "The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals," in *Proc. of ISMIR 2006*, pp. 324–327, 2006.
[20] B. Li, J. A. Burgoyne and I. Fujinaga, "Extending Audacity as a grouth-truth annotation tool," in *Proc. of ISMIR 2006*, pp. 379–380, 2006.
[21] X. Amatriain *et al.*, "The CLAM annotator: A cross-platform audio descriptors editing tool," in *Proc. of ISMIR 2005*, pp. 426–429, 2005.
[22] P. Herrera *et al.*, "MUCOSA: A music content semantic annotator," in *Proc. of ISMIR 2005*, pp. 77–83, 2005.
[23] L. von Ahn, "Games with a purpose," *IEEE Computer Magazine*, vol. 39, pp. 92–94, June 2006.
[24] M. Goto, J. Ogata and K. Eto, "PodCastle: A Web 2.0 approach to speech recognition research," in *Proc. of Interspeech 2007*, 2007.
[25] M. Goto and J. Ogata, "PodCastle: Recent advances of a spoken document retrieval service improved by anonymous user contributions," in *Proc. of Interspeech 2011*, 2011.