AUTOMATIC SET LIST IDENTIFICATION AND SONG SEGMENTATION FOR FULL-LENGTH CONCERT VIDEOS

Ju-Chiang Wang[1,2], **Ming-Chi Yen**[1], **Yi-Hsuan Yang**[1], **and Hsin-Min Wang**[1]
[1]Academia Sinica, Taipei, Taiwan
[2]University of California, San Diego, CA, USA
asriver.wang@gmail.com; {ymchiqq, yang, whm}@iis.sinica.edu.tw

ABSTRACT

Recently, plenty of full-length concert videos have become available on video-sharing websites such as YouTube. As each video generally contains multiple songs, natural questions that arise include "what is the set list?" and "when does each song begin and end?" Indeed, many full concert videos on YouTube contain song lists and timecodes contributed by uploaders and viewers. However, newly uploaded content and videos of lesser-known artists typically lack this metadata. Manually labeling such metadata would be labor-intensive, and thus an automated solution is desirable. In this paper, we define a novel research problem, *automatic set list segmentation of full concert videos*, which calls for techniques in music information retrieval (MIR) such as audio fingerprinting, cover song identification, musical event detection, music alignment, and structural segmentation. Moreover, we propose a greedy approach that sequentially identifies a song from a database of studio versions and simultaneously estimates its probable boundaries in the concert. We conduct preliminary evaluations on a collection of 20 full concerts and 1,152 studio tracks. Our result demonstrates the effectiveness of the proposed greedy algorithm.

1. INTRODUCTION

In recent years, the practice of sharing and watching concert/performance footage on video sharing websites such as YouTube has grown significantly [12]. In particular, we have noticed that many concert videos consist of full-length, unabridged footage, featuring multiple songs. For example, the query "full concert" on YouTube returns a list of more than 2 million relevant videos. Before watching a full concert video, a viewer might like to know if the artist has performed the viewer's favorite songs, and when are those song played in the video. Additionally, after watching a concert video, a viewer may want to know the song titles in order to locate the studio version.

To satisfy such a demand, the uploader or some viewers often post the "set list" with the timecode for each song, [1] so that other viewers can easily fast-forward to the desired song. This metadata can help viewers to navigate a long concert. From a technical point of view, it also helps to extract the live version of a song to enrich a music database. Such a database could be used to analyze performance style, to discover song transition [17], to train classifiers for visual event detection [28], or to generate multi-camera mashups and summaries of concert videos [22,27].

However, newly uploaded videos and those performed by less known artists typically lack this metadata, because manually identifying songs and song segmentation can be time consuming even for an expert. One reason for this is because live performances can differ substantially from the studio recordings. Another reason is that live performances often contain covers of songs by other artists. Even if the annotator can readily identify all songs, it is still necessary to go through the entire video to locate the precise times that each song begins and ends. Therefore, an automated method is desirable to annotate the rapidly growing volume of full-length concert videos available online.

The aim of this paper is threefold. First, we define a novel research problem, i.e. automatic set list segmentation of full concert videos, and discuss its challenges. Second, we propose a greedy approach to tackle the problem. Third, we construct a novel dataset designed for this task and suggest several evaluation methods.

1.1 Task Definition and Challenges

There are two sub-tasks for this research problem: *set list identification* and *song segmentation*. Given a full concert video, the former is to identify the sequence of song titles played in the concert based on a large collection of studio version tracks, assuming that no prior knowledge on the live performance of the artist(s) of the concert is available. The latter task is to estimate the boundaries of each identified song in the set list. This problem poses some interesting challenges as follows:

- A live song can be played in many different ways, e.g., by changing its timbre, tempo, pitch and structure, comparing to the corresponding studio version.

[1] A set list refers to a list of songs that a band/artist has played in a concert, and the timecode corresponds to the starting time of a song. Here is an example of full concert video with set list and timecodes on YouTube: https://www.youtube.com/watch?v=qTOjiniIltQ

Therefore, certain robustness should be considered.

- Live performances often feature transitions between consecutive songs, or even repeated oscillations between the sections of different songs, suggesting that one should identify songs on a small temporal scale.
- Concerts often feature sections with no reference in the collection of studio versions, such as intros, outros, solos, banter, transitions between songs, big rock endings, and applause, amongst others. Unexpected events such as broken instruments, sound system malfunctions, and interrupted songs can also be found. An ideal system should identify them or mark them as unknown songs/events, avoiding including them in a segmented song when appropriate.
- The artist may play cover songs from other artists partially or entirely throughout the concert, resulting in a much larger search space in the music database.
- The audio quality of user-contributed concert videos can vary significantly due to recording factors such as acoustic environment, position, device and user expertise [14]. The quality degradation can amplify the difficulty of the problem.

To tackle the above challenges, one may consider techniques for several fundamental problems in music information retrieval (MIR), such as audio fingerprinting/matching [3, 7], cover song identification [5, 24], audio quality assessment [14], musical event detection/tracking [32, 33], and music signal alignment and segmentation [18]. Therefore, automatic set list segmentation of full concert videos may present a new opportunity for MIR researchers to link music/audio technology to real-world applications.

1.2 Technical Contribution

Our technical contribution lies in the development of a greedy approach that incorporates three components: segmentation, song identification, and alignment (see Section 3). This approach provides a basic view as a baseline towards future advance. Starting from the beginning of the concert, our approach first identifies the candidate songs for a "probe excerpt" of the concert based on segmented music signals. Then, it estimates the most likely song title and boundaries of the probe excerpt based on dynamic time warping (DTW) [18]. This sequential process is repeated until the entire concert video has been processed. To evaluate the proposed algorithm, we collect 20 full concerts and 1,152 studio tracks from 10 artists (see Section 4). Moreover, we suggest three performance metrics for this task (see Section 5). Finally, we demonstrate the effectiveness of the proposed approach and observe that cover song identification works much better than audio fingerprinting for identifying the songs in a live performance (see Section 5).

2. RELATED WORK

According to a recent user study, YouTube was the second most preferred online music streaming service by users in 2012, just behind Pandora [12]. These community-contributed concert videos have been extensively studied in the multimedia community. Most existing works focus on handling the visual content of the concert videos [1, 10, 22, 27, 28]. Relatively little attention, however, has been paid in the MIR community to study the audio content of this type of data. Related work mainly focused on low-level audio signal processing for tasks such as audio fingerprint-based synchronization and alignment for concert video organization [9, 11, 29], and audio quality ranking for online concert videos [14]. More recently, Rafii *et al.* proposed a robust audio fingerprinting system to identify a live music fragment [23], without exploring full-length concert videos and song segmentation. To gain deeper understanding of the content and context of live performance, our work represents an early attempt to use the full concert video data.

We note that our work is also related to PHENICX [6], an ongoing project which aims at enriching the user experience of watching classical music concerts via state-of-the-art multimedia and Internet technologies. With a system for automatic set list segmentation of full concert videos, one could index a large amount of online musical content, extracting information that helps link live performance to the associated video content.

Aside from potential applications, the technical development of our work is highly motivated by several signal matching-based music retrieval problems, which can be categorized into audio fingerprinting (AF) [3, 30], audio matching [21], and cover song identification (CSID) [5, 24], according to their *specificities* and *granularity* [4, 7]. An AF system retrieves the exact audio piece that is the source of a query audio fragment. Audio matching is defined as the task of retrieving from a database all the audio fragments that are musically relevant to a query fragment. In contrast, CSID aims at identifying different renditions of a music piece in the track level (instead of fragment-level). Unlike AF which usually holds robustness to any noises that may apply on the same rendition of a song, audio matching and CSID should handle the musically motivated variations occurring in different performances or arrangements of a music piece [7].

3. PROPOSED GREEDY APPROACH

The proposed approach is outlined in Algorithm 1. It employs an intuitive greedy strategy that recursively probes an excerpt X from the beginning of the unprocessed concert Z, identifies K song candidates ($K = 5$) from the studio database $\mathcal{D}$, selects the most probable song title $s^\star$, estimates the boundaries (i, j) of $s^\star$ in X, and finally removes $s^\star$ from $\mathcal{D}$ and $X(1:j)$ from Z. The process stops when the unprocessed portion of the input concert is shorter than a pre-defined threshold τ. We make the following assumptions while developing Algorithm 1: 1) the performer plays nearly the entire part of a song rather than a certain small portion of the song, 2) a song in the studio database is performed at most once in a concert, and 3) the concert contains only songs from the same artist without covers. In practice, the artist of a concert can be easily known from the video title. Therefore, we only take the studio tracks of the artist to construct $\mathcal{D}$. More details are given below.

Algorithm 1: Set list identification & segmentation

Input: A concert Z; studio track database $\mathcal{D}$; probe length l; end length τ; candidate number K;
Output: Song list $\mathcal{S}$; boundary set $\mathcal{B}$;

$\mathcal{S} \leftarrow \emptyset$; $\mathcal{B} \leftarrow \emptyset$;
while *length*$(Z) > \tau$ **do**
  $X \leftarrow Z(1:l)$, if $l >$length(Z), $l =$ length(Z);
  $\{s_k\}_{k=1}^{K} \leftarrow$ identify the K most probable songs that match X, based on the thumbnails of $\mathcal{D}$;
  $\{s^\star, (i,j)\} \leftarrow$ select the best song from $\{s_k\}_{k=1}^{K}$ and estimate its boundaries on X, based on the complete track of $\mathcal{D}$;
  $\mathcal{S} \leftarrow \mathcal{S} + s^\star$; $\mathcal{B} \leftarrow \mathcal{B} + (i,j)$;
  $\mathcal{D} \leftarrow \mathcal{D} - s^\star$; $Z \leftarrow Z - X(1:j)$;
end

3.1 Segmentation

In our original design, we adopt music segmentation techniques to pre-process both the concert and every studio track in the database. This enhances the robustness to variation of song structure for the music matching and identification processes. However, operating on fine-grained segments of the concert significantly increases the computational time of the algorithm. Therefore, we make heuristic modifications to gain more efficiency as follows.

First, we segment a sufficiently long probe excerpt from the beginning of an unprocessed concert that could include the first entire song played in the unprocessed concert, without involving any musically motivated segmentation. Ideally, we hope the probe length l is longer than the exact song $s^\star$ plus the events prior to $s^\star$ (e.g., banter, applause). In the experiment, we will compare different settings of $l = \alpha \times \mu$, where α is the parameter and μ the mean length of all studio tracks in the database.

Second, each studio track in the database is represented by its thumbnail for better efficiency in the later song candidate identification stage. Similar idea has been introduced by Grosche *et al.* [8]. We develop a simple method analogous to [15] based on structural segmentation. Segmentino [2, 16] is utilized to discover the musically homogeneous sections marked by structure labels such as 'A,' 'B,' and 'N' for each studio track. We compute a weighted factor γ that jointly considers the repetition count and average segment length for each label. The longest segment of the label that has the largest γ is selected as the thumbnail.

3.2 Song Candidate Identification

Song candidate identification uses the probe excerpt as a query and ranks the studio thumbnails in the database. We employ two strategies for the identifier: audio fingerprinting (AF) and cover song identification (CSID). For simplicity, we employ existing AF and CSID methods in this work. For future work, it might be more interesting to integrate the identifier with the subsequent boundary estimator.

For AF, we implement the identifier using the widely-known landmark-based approach proposed in [31]. It extracts prominent peaks (a.k.a. *landmarks*) from the magnitude spectrogram of a reference track (e.g. a studio version) and characterizes each pair of landmarks by the frequencies of the landmarks and the time in between them, which provide indices to a hash table that allows fast retrieval of similarity information [30]. For a query (e.g. a probe excerpt), we see whether there are sufficient number of matched landmarks between the query and a reference track by looking up the hash table. If the query track is a noisy version of the reference track, this approach is likely to perform fairly well, because the landmarks are most likely to be preserved in noise and distortion.

For CSID, we implement the identifier mainly based on the *chroma DCT-reduced log pitch* (CRP) features [19] and the *cross recurrence quantification* (CRQ) approach [25], which correspond to two major components in a state-of-the-art CSID system [26]. Specifically, we first extract the frame-based CRP features for the probe excerpt and each studio track by the Chroma Toolbox [20]. Then, we determine the key transposition using the optimal transposition index (OTI) [25]. To apply CRQ, we follow the standard procedures [25], including constructing the delay coordinate state space vectors, computing the cross recurrence plot, deriving the $Q_{\max}$ score, and performing normalization on the scores across the database. This CSID system (cf. CYWW1) has led to performance comparable to the state-of-the-art systems in the MIREX audio cover song identification task (e.g., on Sapp's Mazurka Collection). [2]

3.3 Song Selection and Boundary Estimation

The next step is to select the most probable song $k^\star$ from the top K studio song candidates, $\{Y_k\}_{k=1}^{K}$, and at the same time estimate its boundaries on the probe excerpt X. Accordingly, our goal is to find a Y_k and the corresponding subsequence $X^\star = X(i^\star : j^\star)$ that results in the best matching between Y_k and $X^\star$, where $1 \leq i^\star < j^\star \leq N$. Such process is based on the DTW alignment between X and each Y_k, as presented in Algorithm 2.

Let $X = \{x_1, \ldots, x_N\}$ and denote the complete track of Y_k as $Y' = \{y_1, \ldots, y_M\}$, where x_i and y_i represent the frame-based CRP vectors and $N > M$. We compute the cost by the negative cosine similarity of CRP between two frames after the OTI key transposition. One can observe that Algorithm 2 includes two sub-procedures of one-side boundary estimation (cf. Algorithm 3). It first executes Algorithm 3 to search for the end boundary j' on X and then *reverses* the search from j' for the start boundary i' using Algorithm 3 with the cost matrix rotated by 180 degrees. We follow the standard procedure to compute the accumulated cost matrix D in [18]. Then, Algorithm 3 searches from $D(\frac{N}{2}+1, M)$ to $D(N, M)$ for the minimum *average cost* of DTW alignments, denoted by $\delta_k^\star$, where the average cost is defined as the accumulated cost divided by the length of its optimal warping path (OWP). The frame index of $\delta_k^\star$ is set as the boundary.

After the K candidates are processed, we pick the one

[2] http://www.music-ir.org/mirex/wiki/2013:Audio_Cover_Song_Identification_Results

Algorithm 2: Boundaries & average cost estimation

Input: Concert excerpt X; a studio track Y';
Output: Boundary pair (i', j'); average cost δ;
$C \leftarrow N$-by-M cost matrix between X and Y';
$(j', \emptyset) \leftarrow$ one-side boundary estimation on C;
$C \leftarrow$ rotate $C(1 : j', 1 : M)$ by 180 degrees;
$(index, \delta) \leftarrow$ one-side boundary estimation on C;
$i' \leftarrow j' - index + 1$;

Algorithm 3: One-side boundary estimation

Input: Cost matrix C;
Output: Boundary β; average cost δ;
$D \leftarrow$ accumulated cost matrix from $C(1, 1)$;
for $1 \leftarrow i$ **to** $\frac{N}{2}$ **do**
  $p^\star \leftarrow$ compute the OWP of $D(1 : \frac{N}{2} + i, 1 : M)$;
  $\Delta(i) \leftarrow D(\frac{N}{2} + i, M)/\text{length}(p^\star)$;
end
$(\delta, index) \leftarrow$ the minimum value and its index of Δ;
$\beta \leftarrow index + \frac{N}{2}$;

with the lowest average cost, $k^\star = \arg\min_k \{\delta_k\}_{k=1}^K$, and set the boundary pair as $(i'_{k^\star}, j'_{k^\star})$. In other words, we re-rank the top K candidates according to the results of Algorithm 2, based on the content of the complete studio tracks.

4. DATA COLLECTION

We collect 20 popular full concert videos (from the first few responses to the query "full concert" to Youtube) and the associated set lists and timecodes from YouTube. Therefore, the music genre is dominated by pop/rock. We manually label the start and end boundaries of each song based on the timecodes, as a timecode typically corresponds to the start time of a song and may not be always accurate. There are 10 artists. For each artist, we collect as many studio tracks as possible including the songs performed in the collected concerts to form the studio database. On average, we have 115.2 studio version tracks for each artist, and each full concert video contains 16.2 live version tracks. Table 1 summarizes the dataset.

5. EVALUATION

5.1 Pilot Study on Set List Identification

We conduct a pilot study to investigate which strategy (i.e., AF or CSID) performs better for set list identification, assuming that the song segmentation is perfect. For simplicity, we extract all the songs from the concert videos according to the manually labeled boundaries and treat each entire live song as a query (instead of thumbnail). We use mean average precision (MAP) and precision@1 with respect to the studio database as the performance metrics. We also perform random permutation ten times for each query to generate a lower bound performance, denoted by 'Random.' One can observe from Table 2 that CSID performs significantly better than AF in our evaluation, showing that the landmark-based AF approach does not work well for live version identification. This confirms our intuition as live rendition can be thought of as a cover version of the studio version [5]. In consequence, we use CSID as the song candidate identifier in the following experiments.

ID	Artist Name	Concerts	Studio Tracks
1	Coldplay	2	96
2	Maroon 5	3	62
3	Linkin' Park	4	68
4	Muse	2	100
5	Green Day	2	184
6	Guns N' Roses	2	75
7	Metallica	1	136
8	Bon Jovi	1	205
9	The Cranberries	2	100
10	Placebo	1	126

Table 1. The full concert dataset.

Method	MAP	Precision@1
AF	0.060	0.048
CSID	**0.915**	**0.904**
Random	0.046	0.009

Table 2. Result for live song identification.

5.2 Performance Metrics

We use the following performance metrics for set list identification and song segmentation: *edit distance* (ED), *boundary deviation* (BD), and *frame accuracy* (FA). The first metric ED is originally used to estimate the dissimilarity of two strings and has been adopted in numerous MIR tasks [13]. We compute the ED between an output song sequence (a list of song indices) and the ground truth counterpart via dynamic programming. The weights for insertion, deletion, and substitution are all set to 1. ED can only evaluate the accuracy of set list identification.

The second metric BD directly measures the absolute deviation in second between the estimated boundary and that of the ground truth for only each correctly identified song, ignoring those wrongly inserted songs in the output set list, as they are not presented in the ground truth. Therefore, the average BD of a concert reflects the accuracy of song segmentation but not set list identification.

The last metric, FA, which has been used in tasks such as melody extraction, represents the accuracy at the frame-level (using non-overlapped frame with length 200 ms). Throughout the concert, we mark the frames between the start and end boundaries of each song by its song index and otherwise by 'x' (belonging to no song). Then, we calculate the percentage of correct frames (the intersection rate) by comparing the output frame sequence with the ground truth counterpart. Therefore, FA can reflect the accuracy of both set list identification and song segmentation.

5.3 Baseline Approach

To study the effectiveness of the song selection and boundary estimation algorithms (see Section 3.3), we construct a baseline approach using Algorithm 1 without Algorithms 2 and 3. Specifically, we select the song $s^\star$ with the largest

ID	A	SG	SO	ED$^\flat$	sBD$^\flat$	eBD$^\flat$	FA
1	7	20	15	17	6.5	89.1	0.317
2	3	17	17	4	3.3	12.3	0.786
3	1	15	15	3	27.2	33.2	0.744
4	8	23	25	14	8.8	66.8	0.441
5	10	19	18	5	11.5	27.8	0.641
6	6	10	11	1	19.1	22.8	0.875
7	2	10	10	6	28.2	39.1	0.428
8	3	22	22	9	28.2	39.6	0.610
9	6	20	21	7	30.7	35.9	0.653
10	9	17	15	4	5.3	9.8	0.758
11	9	22	21	3	6	8.7	0.860
12	4	17	19	7	32.0	21.9	0.681
13	2	9	12	5	110	155	0.509
14	1	17	17	2	20.1	18.4	0.777
15	2	11	11	7	50.9	72.9	0.393
16	3	17	20	9	36.9	24.7	0.544
17	4	13	11	4	48.1	94.3	0.626
18	3	23	22	10	10	34.8	0.636
19	5	7	7	3	42.4	13.6	0.584
20	5	15	13	9	42.4	36.6	0.465
AVG(α=1.5)		16.2	16.1	6.5	23.4	42.9	0.616
AVG(α=1.2)		16.2	18	7.3	25.7	57.3	0.582
AVG(α=1.8)		16.2	14.6	8.4	29.3	45.3	0.526
Baseline		16.2	19.7	8.9	229	241	0.434

Table 3. Result of the greedy approach with α=1.5 for the 20 full concerts and their average (AVG) performance. While 'AVG (α=1.2 or α=1.8)' only shows the average performance with different l settings. 'Baseline' represents the average performance of the approach in Section 5.3. Additional abbreviations: A (Artist ID), SG (number of Songs in the Ground truth set list), SO (number of Songs in the Output set list), sBD (start BD), and eBD (end BD). Symbol $\flat$ marks the metrics that are the smaller the better.

CSID score on a probe excerpt. The start boundary is the start point of the probe excerpt, and the end boundary is the length($s^\star$). Then, we begin the next probe excerpt on a hop of $0.1\times$length($s^\star$).

5.4 Result and Discussion

Table 3 shows the quantitative result of each concert, the average performance (AVG) with different values of l, and the average performance of Baseline. Figure 1 depicts the qualitative results of three concerts, including the best, medium, and the worst cases according to FA in Table 3.

The following observations can be made. First, the AVG performances of the complete approach are significantly better than those of Baseline in all metrics, demonstrating the effectiveness of Algorithms 2 and 3. Second, further comparison among AVG performances with different l settings shows that α=1.5 performs the best, revealing that live versions are likely longer than studio ones, but overly large l could yield more deletions, as observed by the smaller SO of α=1.8. Third, on average our approach gives similar number of songs of a concert as that of ground truth (16.1 versus 16.2). Fourth, we find an interesting linkage between the result and the style of the live performance. For example, we find that our approach performed poorly for 'Maroon 5' (A=2) and 'Metallica' (A=7). As can be observed from the last two rows of Figure 1, Maroon 5 tends to introduce several non-song sections such as jam and banter, which cannot be accurately modeled by our approach. They also like to make the live renditions different from their studio versions. On the other hand, we conjecture that the riffs in the heavy metal music such as Metallica may be the main reason degrading the performance of matching thumbnails by CSID, because such riffs lack long-term harmonic progressions. Fifth, the performance for 'Bon Jovi' (A=8) is poor, possibly because of the relatively large quantity of studio tracks in the search space. Finally, owing to possible big rock endings or repetitive chorus in the live performance, our approach relatively cannot estimate accurate end boundary of the songs in a concert, as reflected by larger eBD than sBD. Our approach sometimes insert songs that are relatively short in length, as can be observed in Figure 1. The above two observations suggest that advanced methods (over Algorithm 3) for boundary estimation and regularizing the song length might be needed.

In short, while there is still much room for improvement, we find that the result of the proposed greedy approach is quite satisfactory in some cases (e.g., Concert 6 in Figure 1). The greedy approach is preliminary in nature. We believe that better result can be obtained by explicitly addressing the challenges described in Section 1.1.

6. CONCLUSION AND FUTURE DIRECTION

In this paper, we have proposed a novel MIR research problem with a new dataset and a new greedy approach to address the problem. We have also validated the effectiveness of the proposed approach via both quantitative and qualitative results. We are currently expanding the size of the dataset and conducting more in-depth signal-level analysis of the dataset. Due to the copyright issue on the studio track collection, however, it is not likely to distribute the dataset. We will propose this task to MIREX to call for more advanced approaches to tackle this problem.

7. ACKNOWLEDGEMENT

This work was supported by Academia Sinica–UCSD Postdoctoral Fellowship to Ju-Chiang Wang, and the Ministry of Science and Technology of Taiwan under Grants NSC 101-2221-E-001-019-MY3 and 102-2221-E-001-004-MY3.

8. REFERENCES

[1] A. Bagri, F. Thudor, A. Ozerov, and P. Hellier. A scalable framework for joint clustering and synchronizing multi-camera videos. In *Proc. EUSIPCO*, 2013.

[2] C. Cannam et al. MIREX 2013 entry: Vamp plugins from the centre for digital music. In *MIREX*, 2013.

[3] P. Cano et al. A review of audio fingerprinting. *J. Sign. Process. Syst.*, 41(3):271–284, 2005.

[4] M. A. Casey et al. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[5] D. PW Ellis and G. E Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proc. ICASSP*, pages IV–1429, 2007.

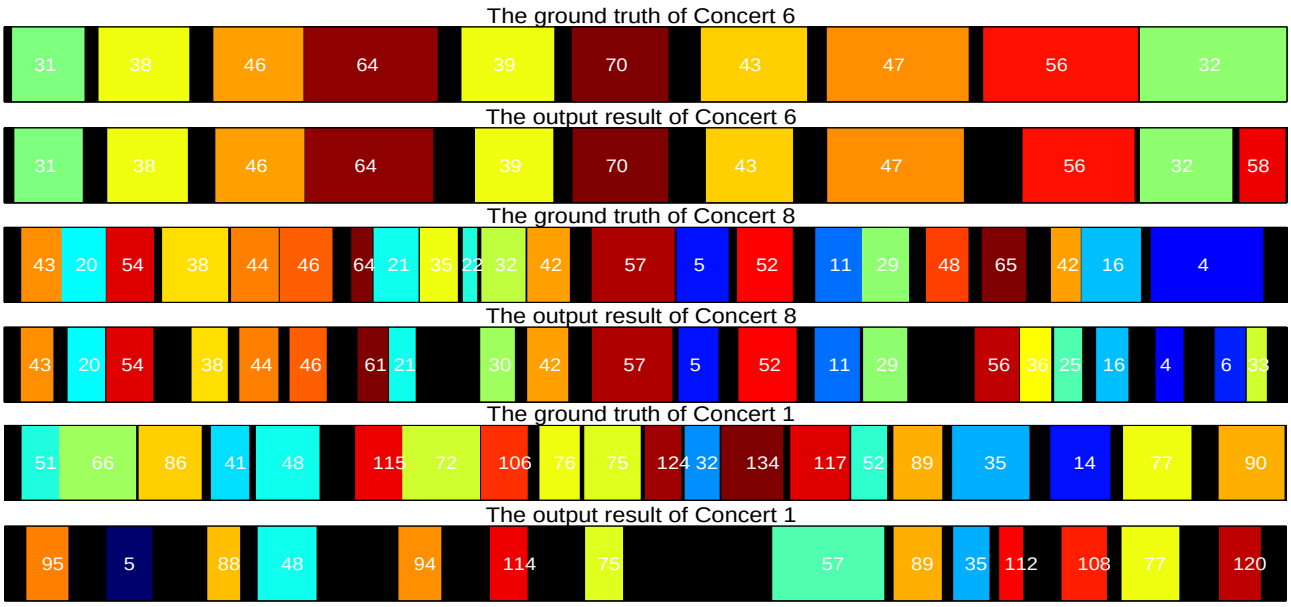

Figure 1. Qualitative result of three concerts, which represent the best (Concert 6, 'Guns N' Roses'), medium (Concert 8, 'Linkin' Park'), and worst (Concert 1, 'Metallica') output cases in the dataset. Black blocks correspond to no song. Different songs are marked by different colors. The number in a song block stands for the song index in the studio database. Note that Song 42 ('Numb') was sung twice in Concert 8, firstly by 'Linkin' Park' and then by 'featuring Jay-Z.'

[6] E. Gómez et al. PHENICX: Performances as highly enriched and interactive concert experiences. In *Proc. SMC*, 2013.

[7] P. Grosche, M. Müller, and J. Serrà. Audio content-based music retrieval. *Multimodal Music Processing*, 3:157–174, 2012.

[8] P. Grosche, M. Müller, and J. Serrà. Towards cover group thumbnailing. In *Proc. ACM MM*, pages 613–616, 2013.

[9] M. Guggenberger, M. Lux, and L. Boszormenyi. AudioAlign - Synchronization of A/V-streams based on audio data. In *Proc. IEEE ISM*, pages 382–383, 2012.

[10] L. Guimarães et al. Creating personalized memories from social events: community-based support for multi-camera recordings of school concerts. In *Proc. ACM MM*, 2011.

[11] L. Kennedy and M. Naaman. Less talk, more rock: Automated organization of community-contributed collections of concert videos. In *Proc. WWW*, pages 311–320, 2009.

[12] J. H. Lee and N. M. Waterman. Understanding user requirements for music information services. In *Proc. ISMIR*, 2012.

[13] Kjell Lemström. *String matching techniques for music retrieval*. Ph.D. Thesis, University of Helsinki, 2000.

[14] Z. Li, J.-C. Wang, J. Cai, Z. Duan, H.-M. Wang, and Y. Wang. Non-reference audio quality assessment for online live music recordings. In *Proc. ACM MM*, pages 63–72, 2013.

[15] B. Martin, P. Hanna, M. Robine, and P. Ferraro. Indexing musical pieces using their major repetition. In *Proc. JCDL*, pages 153–156, 2011.

[16] M. Mauch, K. Noland, and S. Dixon. Using musical structure to enhance automatic chord transcription. In *Proc. ISMIR*, pages 231–236, 2009.

[17] B. McFee and G. Lanckriet. The natural language of playlists. In *Proc. ISMIR*, pages 537–542, 2011.

[18] M. Müller. *Information retrieval for music and motion*. 2007.

[19] M. Muller and S. Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Trans. Audio, Speech, and Lang. Process.*, 18(3):649–662, 2010.

[20] M. Müller and S. Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *Proc. ISMIR*, pages 215–220, 2011.

[21] M. Müller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proc. ISMIR*, 2005.

[22] S. U. Naci and A. Hanjalic. Intelligent browsing of concert videos. In *Proc. ACM MM*, pages 150–151, 2007.

[23] Z. Rafii, B. Coover, and J. Han. An audio fingerprinting system for live version identification using image processing techniques. In *Proc. ICASSP*, pages 644–648, 2014.

[24] J. Serra, E. Gómez, and P. Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In *Advances in Music Information Retrieval*, pages 307–332. 2010.

[25] J. Serra, X. Serra, and R. G Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):093017, 2009.

[26] J. Serrà, M. Zanin, and R. G Andrzejak. Cover song retrieval by cross recurrence quantification and unsupervised set detection. In *MIREX*, 2009.

[27] P. Shrestha et al. Automatic mashup generation from multiple-camera concert recordings. In *Proc. ACM MM*, pages 541–550, 2010.

[28] C. GM Snoek et al. The role of visual content and style for concert video indexing. In *Proc. ICME*, 2007.

[29] K. Su, M. Naaman, A. Gurjar, M. Patel, and D. PW Ellis. Making a scene: alignment of complete sets of clips based on pairwise audio match. In *Proc. ICMR*, page 26, 2012.

[30] A. Wang. An industrial strength audio search algorithm. In *Proc. ISMIR*, pages 7–13, 2003.

[31] C.-C. Wang, J.-S. R. Jang, and W. Li. Speeding up audio fingerprinting over GPUs. In *Proc. ICALIP*, 2014.

[32] J.-C. Wang, H.-M. Wang, and S.-K. Jeng. Playing with tagging: A real-time tagging music player. In *ICASSP*, 2012.

[33] S.-Y. Wang, J.-C. Wang, Y.-H. Yang, and H.-M. Wang. Towards time-varying music auto-tagging based on CAL500 expansion. In *Proc. ICME*, 2014.