SEGMENTATION-BASED LYRICS-AUDIO ALIGNMENT USING DYNAMIC PROGRAMMING

Kyogu Lee
Media Technology Lab, Gracenote
Emeryville, CA 94608
klee@gracenote.com

Markus Cremer
Media Technology Lab, Gracenote
Emeryville, CA 94608
mcremer@gracenote.com

ABSTRACT

In this paper, we present a system for automatic alignment of textual lyrics with musical audio. Given an input audio signal, structural segmentation is first performed and similar segments are assigned a label by computing the distance between the segment pairs. Using the results of segmentation and hand-labeled paragraphs in lyrics as a pair of input strings, we apply a dynamic programming (DP) algorithm to find the best alignment path between the two strings, achieving segment-to-paragraph synchronization. We demonstrate that the proposed algorithm performs well for various kinds of musical audio.

1 INTRODUCTION

As the market for portable media players increases rapidly, more and more manufacturers and multimedia content providers are searching for outstanding features that will set their offering apart from the rest. At the same time the rush towards online distribution of digital assets has stripped most content of rich metadata such as album artwork and lyrics. While artwork is gaining a lot of traction with the introduction of higher resolution color displays in devices, only few high end devices offer the ability to display lyrics in a readable form along with music. As devices tend to get smaller and smaller with advancing technology, there is not much hope for displays to become large enough for the user to comfortably follow song lyrics without some form of synchronized presentation. Some devices are already capable of displaying lyrics synchronized to the playback of music using manually inserted time stamps into the lyrics file.

However, this approach does not scale well for large collections, so an automated approach to align lyrics with music is the strongly preferable approach. It is notable that word-by-word level synchronization for most applications is not necessary. A paragraph or line-by-line synchronization, whatever is more appropriate for the device display resolution, will be sufficient. As this promises to be a fairly solvable problem, to which a scalable automated solution could be provided, it has recently attracted a number of researchers in the music information retrieval (MIR) community.

Wang *et al*., for example, have proposed a hierarchical approach for automatic alignment of acoustic musical signals with textual lyrics [11, 9]. They decompose the problem into two separate tasks — a higher-level section-based alignment followed by lower-level per-line alignment. To this end, they first process audio to obtain high-level structural information such as Measure, Chorus and Singing Voice Section. In parallel, textural lyrics are analyzed and each section is labeled with one of the pre-defined section types. In this text processing stage, they also compute approximate durations of each section and line. However, their algorithm is limited by strong assumptions about the song structure as well as the fixed rhythmic structure.

A different approach has been taken by Chen *et al*. who have presented an automatic lyrics-audio synchronization system using the low-level acoustic features only [3]. Their algorithm has two main components: 1) vocal/non-vocal detector and 2) alignment of the audio signal with its lyrics at multiple levels using acoustic models. Given a musical audio signal, a vocal/non-vocal classifier detects candidates for the singing voice sections. In parallel, they construct the grammar net from the lyrics and force an alignment utilizing the previously obtained acoustic model units with a maximum likelihood linear regression technique. They have tested their algorithm on a small set of Chinese song segments and achieve a boundary accuracy of 81.5% at the phrase level.

Fujihara *et al*. tackle the lyrics-audio alignment challenge by solving three sub-problems in series [8]: *i.e*., 1) separation of singing voice, 2) singing voice detection, and 3) alignment of segregated vocal signals with lyrics using a *Viterbi*-based matching technique. At the final alignment stage, they first build a language model from the lyrics using only vowel phonemes and short pauses between word, sentence or phrase boundaries. They also employ an adaptation of a phone model to the specific singer of the input audio signal to improve performance. Using 10 Japanese popular songs as test bed, they have achieved over 90% accuracy for eight songs.

The ultimate goal of the lyrics-audio alignment systems described so far is to automate karaoke-style synchronization at a line or word level. Although a word-level or even

syllable-level synchronization may appear to be ideal, it is extremely challenging to achieve and usually involves solving other difficult problems such as singing voice separation or constructing proper speech models.

In this paper, we deviate from this goal and propose a solution to a simpler, more basic problem: we aim to align song lyrics to the corresponding audio signal at a segment-to-paragraph level. Paragraph-level alignment may not be sufficient for karaoke applications, but as stated initially we believe this will help users follow the lyrics as they listen to music by providing them with referential points over time. Furthermore, as reported by Wang *et al*. [11, 9], section- or segment-level alignment provides an initial solution that makes lower-level alignment such as line- or word-level alignment easier and more robust.

This publication is organized as follows. We describe our proposed method for automatic lyrics-audio alignment in detail in the next Section. Thereafter, in Section 3, we present experimental results with several real examples. Finally, in Section 4, we draw conclusions from the previously presented results, and give an outlook on future work in this domain.

2 METHOD

As mentioned in Section 1, our system for lyrics-audio alignment strives to achieve paragraph-to-segment level synchronization. The motivations for this are as follows. First, we find that the segment structure in musical audio corresponds approximately to the paragraphs in lyrics. For example, a `verse` and/or `chorus` section is found in both audio and lyrics for most popular music, even when investigating one without knowledge of the other. Therefore, if we can divide an entire song into the appropriate segments, we can search for the corresponding paragraphs in lyrics.

Second, paragraph-to-segment level alignment is far easier and more robust than word-level or syllable-level alignment because the latter usually depends on other complex algorithms such as singing voice separation and/or speech recognition, which are very challenging problems by themselves. On the other hand, structural music segmentation has achieved fairly good performance with relatively simple and straightforward techniques, as previously discussed by many other researchers [7, 5, 1, 10].

Therefore, the first stage in our system consists of a segmentation of musical audio, and is described in the next paragraph.

2.1 Structural Segmentation

Structural music segmentation is one of the major research topics in the current field of MIR. We do not intend to solve this problem in this paper, however, and thus we briefly describe the algorithm we use for segmentation as an example.

Most contemporary and classical music has a certain structure or pattern with regard to its temporal evolution that contains systematic change and repetition; for example, a pattern like {`intro-verse-chorus-verse-chorus-bridge-solo-chorus-outro`} is observed to be very common in rock or pop music. Structural music segmentation is referred to as finding the boundaries between these sections. Many algorithms have been proposed so far to find the section boundaries in musical audio. We use an approach that is based on the self-similarity matrix of low-level features derived from the audio signal [6, 7].

In this algorithm, the entire song or musical piece is processed using common spectral analysis methods, and a sequence of feature frames is obtained. A 2-dimensional self-similarity matrix S is then computed, where an element $S(i, j)$ represents a similarity measure between the feature-frame pair (f_i, f_j). A cosine distance is used as similarity metric. Then a kernel correlation is performed on the similarity matrix S to generate a 1-dimensional novelty score, whereby peaks indicate significant changes in musical audio. The novelty score is smoothed in order to suppress spurious peaks. Subsequently, the final segment boundaries are obtained by choosing the peaks whose values are above a certain heuristically determined threshold. Foote and Cooper also use the self-similarity matrix and the segmentation results to cluster the segments and summarize music [4, 5], where similar segments are grouped as a *cluster* like `chorus` or `verse`.

The final output of the segmentation algorithm results therefore in a sequence of segment labels and their corresponding time boundaries. For example, using the algorithm for segmentation outlined above, the final output of *"Super Trouper"* by *A*Teens* is {G-B-D-C-A-B-D-C-A-C-A}, where each label or letter represents a segment, and the segments with the same label represent a cluster. At the clustering stage, we locate the cluster within which the segments are closest to each other, and label it as `chorus`, which is denoted as 'A' in the above example. This simplified heuristic is based on the observation that in by large most popular music the `chorus` section remains mostly unaltered throughout the entire song, as it represents the main theme or hook of the song which is emphasized by repetition as much as other musical techniques.

In the following sections, we explain how we align the paragraphs found in lyrics with the segments that our segmentation algorithm was able to retrieve.

2.2 Paragraph-to-Segment Alignment

As is mentioned above, the segmentation algorithm outputs a sequence of labels with additional clustering information. This and the location of `chorus` sections in particular, yield partial but useful information about how to coarsely align lyrics with audio, provided that we have lyrics that have

been hand-labeled at a paragraph or section level. That is, we can synchronize lyrics with audio using one or multiple `chorus` sections as *anchoring* points. For instance, the structure of lyrics in the above example by *A*Teens* is {`intro-verse-verse-chorus-chorus-verse-verse-chorus-chorus-bridge-chorus-chorus-chorus`}. Although there are seven `chorus` paragraphs in lyrics, we can group the consecutive `chorus` paragraphs to obtain three of them, as many as have been derived from the audio segmentation.

However, there are usually many other segments in the audio signal that we don't have any knowledge of as to their correspondence with the appropriate paragraphs within the lyrics. For example, using the above mentioned example again, a segment 'G' could be an instrumental `intro` or an `intro` with some lyrics content. For repetitive segments such as 'B', 'C', or 'D', we can assume them to be a `verse` section because the `verse` theme or themes usually repeat in most popular music. However, 'B', 'C', or 'D' may be a repetitive instrumental section, as well. Therefore, even if we achieve some basic alignment using `chorus` sections as points of reference we still will have many other segments unaligned, about which we have too little information as to their contents. We propose in the next section a simple solution to this alignment problem using a dynamic programming (DP) algorithm.

2.3 Dynamic Programming

In a dynamic programming (DP) algorithm an optimization problem is first divided into smaller sub-problems, and the optimal scores for the solution of every sub-problem are stored and used to solve an entire problem, without resolving the sub-problems over and over again [2]. DP algorithms are widely used in applications such as DNA sequence matching to efficiently and optimally align two strings of different length, and compute the distance between them.

We can use a DP algorithm for our task because, after structural segmentation of audio, we have obtained two sequences of input strings — one derived directly from the audio signal and the other extracted from the structural clues within the lyrics text — whereby both sequences are of different length. In order to find the minimum-cost alignment path using DP, however, we first need to compute an *error matrix* E from the two input sequences, where an element $E(i, j)$ represents the distance between the ith element of a sequence S_1 and the jth element of a sequence S_2. Hence, it is critical to appropriately define the pairwise distance between an audio segment and a lyrics paragraph so as to achieve meaningful alignment results.

Therefore, we first simplify the input string describing the audio structure by reducing the number of label types. That is, we re-label the audio segments so that there are only three labels — 'C' for `chorus`, 'V' for `verse` and 'O' for all others. However, as mentioned above, we don't know for certain which segment within the audio structure corresponds to which in the lyrics except for the `chorus`. Thus we label the segments that repeat twice or more as `verse` or 'V', and those that appear only once as 'others' or 'O'. As to labeling lyrics, we define four different labels — 'C' for `chorus`, 'V' for `verse`, 'I' for `intro` and 'B' for `bridge` — which cover all of the lyrics in our experiment data set.

The next step consists of defining a distance measure between every paragraph-segment pair. First, we determine the minimum distance between a `chorus` pair to use it as an anchor in the alignment process. This is essential because a `chorus` section is the most representative part in most popular music and therefore we would like to avoid misalignment there. Secondly, we assign the second smallest distance to a `verse` pair, but with less confidence than `chorus` pairs, because we are less certain about it being a true `verse` segment from our segmentation results than with the `chorus` segments — these segments can more often be confused with introductory solo parts, bridges or other sections aligned with the verses in one fashion or another. We define the distance between every further pair in a similar manner. Table 1 shows a distance matrix that has been generated as part of our experiments for every possible lyrics-audio pair.

Table 1. Distance matrix between paragraph-segment pairs

audio segments \ lyrics paragraph	C	V	I	B
C	0.0	0.5	0.3	0.5
V	0.5	0.2	0.5	0.3
O	0.7	0.5	0.3	0.3

After the error matrix E is computed using the pairwise distance matrix in Table 1, an accumulated error matrix is calculated by storing the minimum cost of the three possible directions from the previous step (*i.e.*, $\rightarrow, \downarrow, \searrow$). By backtracking, the DP algorithm then retrieves the optimal alignment path with minimum accumulated error.

3 EXPERIMENTS

Using the paragraph-to-segment level alignment algorithm described so far, we have performed experiments on a number of musical items representing a variety of genres and styles of popular music. Figure 1 shows the alignment results of *"Super Trouper"* by *A*Teens*. Shown is the error matrix between the alignment of lyrics and audio at a paragraph-to-segment level, which is computed using the distance measure in Table 1. The minimum-cost alignment path found by the DP algorithm is also displayed in thick, solid lines.

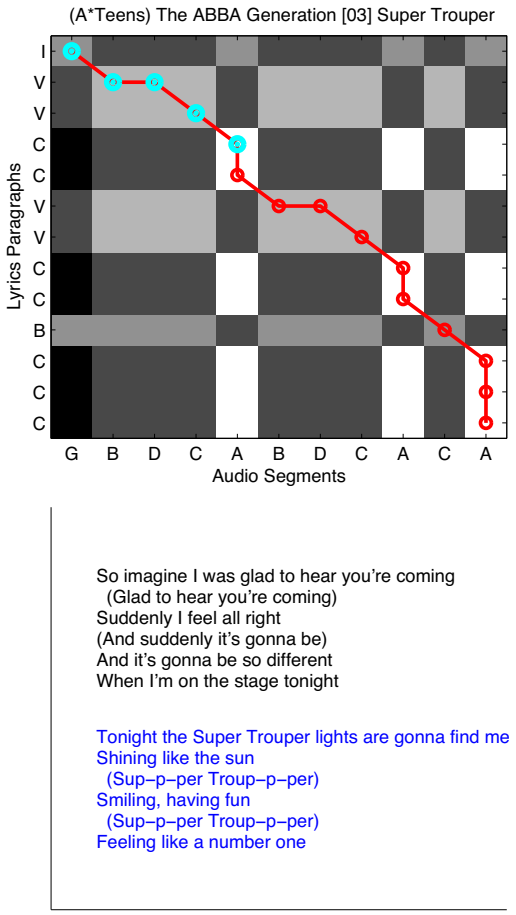

Figure 1. Alignment results of *"Super Trouper"* by *A*Teens*. At the top is shown the error matrix along with the optimal alignment path. At the bottom is displayed lyrics paragraphs corresponding to previous and current audio segments being played.

As shown in Figure 1, `chorus` sections ('C' for lyrics and 'A' for audio) are always aligned since the distance between the two is 0 by definition. Thereby anchoring points are provided which will help to align the entire song subsequently. This also allows for automatic grouping of consecutive `chorus` paragraphs in lyrics and mapping them to corresponding `chorus` segments in audio.

However, some issues can be observed in these alignment results. First of all, we can never guarantee a one-to-one mapping between lyrics paragraphs and audio segments because the number of paragraphs in lyrics seldom match the number of segments in audio. Therefore, there are cases where two or more paragraphs are assigned to one audio segment or vice versa. We approximately resolve the first case by equally dividing the audio segment by the number of lyrics paragraphs so that both lyrics and audio have the same number of paragraphs/segments. For example, we observe in Figure 1 that two `chorus` ('C') paragraphs are assigned to the first `chorus` ('A') segment. Here, we display the first 'C' in the beginning of 'A' and the second 'C' at the half the duration of 'A'. This approach results in a more realistic though still somewhat inaccurate alignment.

The more serious issue occurs when there are more than two audio segments assigned to one lyrics paragraph, such as 'B' and 'D' assigned to the first 'V', as shown in Figure 1. The same solution we use for the first case — *i.e.*, subdividing 'V' into two equal paragraphs and mapping them to 'B' and 'D' — won't work because 'B' is an *instrumental* segment in this example. Therefore, correct mapping would result in {V-V}–{D-C}, and in this case by skipping the instrumental 'B' segment. However, we are unable to conclude from the segmentation results which segment contains the singing voice and which is purely instrumental, or if indeed both contain part of the sung lyrics.

We therefore increase the alignment accuracy by first classifying the audio segments into vocal and non-vocal parts, and then applying the alignment algorithm only on the audio segments that have a higher likelihood of containing vocals. We perform segment-level vocal/non-vocal discrimination using the state-of-the-art classification algorithm. Table 2 shows the vocal/non-vocal classification results on *"Super Trouper"* by *A*Teens*.

Table 2. Vocal/non-vocal classification results

Segment	G	B	D	C	A	B	D	C	A	C	A
V/NV classification	V	NV	V	V	V	NV	V	V	V	V	V

Using the above referenced vocal/non-vocal classification results, we can simplify the problem significantly by ignoring the non-vocal audio segments in the alignment process. That is, we replace the original segmentation with {G-D-C-A-D-C-A-C-A} and apply the DP algorithm to obtain the final alignment results as shown in Figure 2.

As shown in Figure 2, two instrumental audio segments (denoted by 'B') are not assigned any lyrics, and this provides a much increased accuracy for following the lyrics along the audio during playback. We also notice that dividing 'A' segments into sub-segments of equal length similarly leads to more precise alignment (indicated in dash-dot lines).

Our alignment algorithm has been tested on 15 popular music items of different musical styles, including pop, rock, R&B, punk and so on. The number of paragraphs in the lyrics for each song of this set varies from 5 to 27. Table 3 displays more specific information about the songs in the test bed. Because the algorithm proposed in this paper has been designed to obtain paragraph-to-segment alignment, we have evaluated the alignment accuracy by compar-

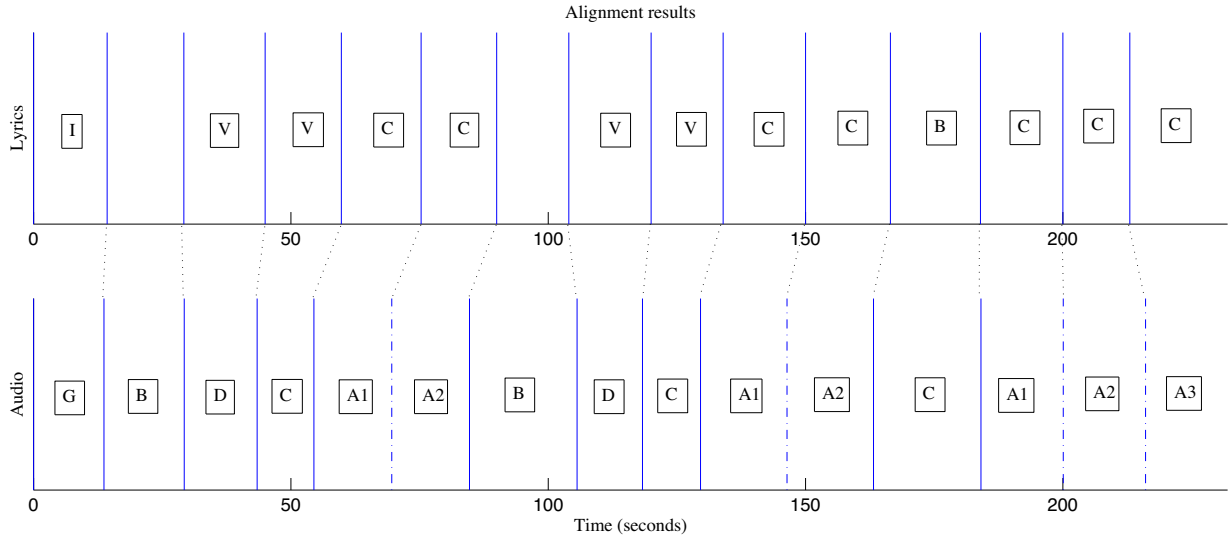

Figure 2. Final alignment results of *"Super Trouper"* by *A*Teens*. `chorus` sections in audio ('A') are sub-divided by dash-dot lines as explained in the text.

ing the manually-marked starting point of each paragraph in the corresponding audio file to that of the automatically selected audio segment.

Using 174 paragraphs in total in the test bed detailed in Table 3, overall we have obtained an average error of 3.50 seconds and a standard deviation of 6.76 seconds. These results reflect the robustness of our alignment algorithm, even though the test songs' styles and lyrical/musical structures vary to a great degree, and the segmentation/clustering results are noticeably different from song to song. The song yielding the best result is *"Hand In My Pocket"* by *Alanis Morissette*, which achieved an average error of 0.91 seconds with a standard deviation of 1.19 seconds. *"I Ran (So Far Away)"* by *A Flock Of Seagulls* produced the least satisfying outcome of 11.25 and 10.48 seconds in average error and standard deviation, respectively.

4 CONCLUSIONS

In this publication, we have presented an algorithm to align song lyrics with a corresponding musical audio recording at a paragraph-to-segment level, as we believe that multiple applications in the entertainment market can benefit from such an alignment. To this avail, we first perform a structural segmentation and clustering of the audio signal, which outputs a sequence of labels where similar segments have been assigned the same label. Lyrics has been hand-labeled at a paragraph level, and we use this pair of label strings as an input to a dynamic programming algorithm. Based on the paragraph-segment distance measure that has been constructed to precisely fit the requirements, the DP algorithm finds the minimum-cost alignment path between lyrics and audio. In addition, we improve the alignment performance by detecting the vocal segments in audio and discarding the non-vocal segments in the alignment process.

Experiments on various kinds of popular music show that the proposed algorithm successfully synchronizes lyrics to audio with all the unintended variations caused by the inaccuracy of the segmentation and clustering of the audio signal.

In the near future, we consider a hierarchical system for lyrics-audio alignment as suggested by Wang *et al*. [11, 9]. In other words, we plan to develop an algorithm for lower-level per-line alignment based on the paragraph-level alignment we achieved in this paper. We believe this hierarchical approach, or an equivalent "divide-and-conquer" method, will be more robust and accurate than performing line-level alignment for an entire song without additional intermediate steps. In addition, a vocal/non-vocal classifier which operates on a finer time grid will allow to more precisely locate the singing voice sections, which we believe will further correct the mis-alignment caused by false segmentation boundaries.

5 REFERENCES

[1] Jean-Julien Aucouturier and Mark Sandler. Segmentation of musical signals using hidden markov models. In *Proceedings of the Audio Engineering Society*, 2001.

[2] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

Table 3. Songs in the test bed

Song ID	Title	Artist	Genre	# of paragraphs	# of segments
1	Hold On Loosely	.38 Special	Rock	9	13
2	I Ran (So Far Away)	A Flock Of Seagulls	Pop	9	12
3	Try Again	Aaliyah	R&B	27	13
4	Don't Turn Around	Ace Of Base	Pop	17	12
5	Hand In My Pocket	Alanis Morissette	Rock	11	10
6	Save Me	American Hi-Fi	Alternative Rock	12	9
7	There's A Star	Ash	Punk	7	10
8	Super Trouper	A*Teens	Pop	13	11
9	Like Dylan In The Movies	Belle & Sebastian	Indie Rock	7	12
10	Summer House	Better Than Ezra	Alternative Rock	5	9
11	Crazy In Love	Beyoncé Feat. Jay-Z	R&B	24	9
12	White Wedding	Billy Idol	Pop	13	11
13	That's Cool	Blue County	Country & Folk	7	10
14	Parklife	Blur	Rock	10	7
15	Here Comes My Baby	Cat Stevens	Pop	7	6

[3] Kai Chen, Sheng Gao, Yongwei Zhu, and Qibin Sun. Popular song and lyrics synchronization and its application to music information retrieval. In *Proceedings of SPIE*, 2006.

[4] Matthew Cooper and Jonathan Foote. Automatic music summarization via similarity analysis. In *Proceedings of the International Conference on Music Information Retrieval*, Paris, France, 2002.

[5] Matthew Cooper and Jonathan Foote. Summarizing popular music via structural similarity analysis. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2003.

[6] Jonathan Foote. Visualizing music and audio using self-similarity. In *Proceedings of ACM International Conference on Multimedia*, Orlando, Florida, 1999.

[7] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of International Conference on Multimedia and Expo*, New York, NY, 2000.

[8] Hiromasa Fujihara, Masataka Goto, Jun Ogata, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Automatic synchronization between lyrics and music cd recordings based on viterbi alignment of segregated vocal signals. In *Proceedings of IEEE International Symposium on Multimedia*, San Diego, CA, 2006.

[9] Min-Yen Kan, Ye Wang, Denny Iskandar, Tin Lay Nwe, and Arun Shenoy. Lyrically: Automatic synchronization of textual lyrics to acoustic music signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 16(2):338–349, 2008.

[10] Mark Levy and Mark Sandler. New methods in structural segmentation of musical audio. In *Proceedings of European Signal Processing Conference*, Florence, Italy, 2006.

[11] Ye Wang, Min-Yen Kan, Tin Lay Nwe, Arun Shenoy, and Jun Yin. Lyrically: Automatic synchronization of acoustic musical signals and textual lyrics. In *Proceedings of ACM Conference on Multimedia*, pages 212–219, New York, NY, 2004.