

# Aligning Meeting Recordings Via Adaptive Fingerprinting

TJ Tsai<sup>1,2</sup>, Andreas Stolcke<sup>3</sup>

<sup>1</sup>EECS Department, University of California Berkeley, Berkeley, CA USA

<sup>2</sup>International Computer Science Institute, Berkeley, CA USA

<sup>3</sup>Microsoft Research, Mountain View, CA USA

tjtsai@eecs.berkeley.edu, stolcke@icsi.berkeley.edu

## Abstract

This paper proposes a robust and efficient way to temporally align a set of unsynchronized meeting recordings, such as might be collected by participants' cell phones. We propose an adaptive audio fingerprint which is learned on-the-fly in a completely unsupervised manner to adapt to the characteristics of a given set of unaligned recordings. The design of the adaptive audio fingerprint is formulated as a series of optimization problems which can be solved very efficiently using eigenvector routines. We also propose a method of aligning sets of files which uses the cumulative evidence from previous alignments to help align the weakest matches. Based on challenging alignment scenarios extracted from the ICSI meeting corpus, the proposed alignment system is able to achieve  $> 99\%$  alignment accuracy at a 100 ms error tolerance.

**Index Terms:** alignment, audio fingerprinting, meeting recordings, adaptive

## 1. Introduction

Consider the following scenario. A group of participants have a meeting in a large conference room. Whenever a person arrives at the meeting, they place their cell phone on the table in front of them and use it as a local audio recording device. Let's say person A arrives at time  $t = 0$  minutes and begins recording. Person B arrives at time  $t = 2$ . Person C joins remotely via skype at  $t = 5$ , and he too simply places his cell phone in front of him at his remote location. Person D arrives late at time  $t = 25$  minutes and begins recording. Some people leave the meeting early; others stay late. Some people don't overlap. At the end of the meeting, everyone has an audio recording. We would like to take these unsynchronized, overlapping audio recordings and generate a single high-quality "summary" recording of the entire meeting. This paper proposes a method for accomplishing this in an efficient and robust manner.

One main problem that must be solved is to align the files in time. Figure 1 shows a graphical depiction of this problem. Once the files are temporally aligned, we can generate a "summary" recording by simply averaging the available channels or applying a beamforming method. Using a simple cross-correlation approach to align the recordings would be prohibitively expensive in this situation, where the audio recordings are very long and the offsets might be on the order of 25 minutes, as in the example above. For example, given 10 recordings that are each 1 hour long and sampled at 8kHz, a simple pairwise cross-correlation approach that considers all possible offsets would require about  $3.7 \times 10^{16}$  multiplications. Using the file timestamps may not be a reliable way to align files. The timestamps on the files might be corrupted or un-

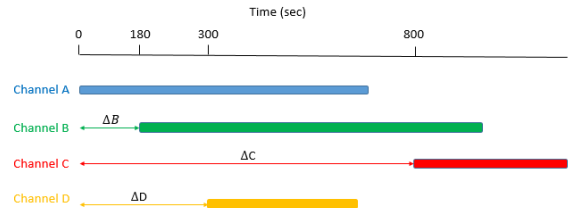


Figure 1: Graphical depiction of an alignment scenario.

available, they might be modified in the process of copying files between devices or compressing to mp3, or they might simply not be accurate enough – the clocks on the recording devices may be skewed or the file timestamps might have limited precision (e.g. only 2 seconds for timestamps on FAT drives).

Several works have explored the synchronization of consumer videos of the same live event using audio information. Most of these works apply an out-of-the-box audio fingerprinting method to the videos in a pairwise manner. For example, Shrestha et al. [1] apply the Philips fingerprint [2] to synchronize live video recordings of the same event in a pairwise manner. The Philips approach considers 33 logarithmically-spaced bands below 2 kHz and performs 32 comparisons at each frame, where each comparison considers whether the energy difference in adjacent frequency bands increases or decreases in 2 consecutive frames. The 32 comparisons are represented compactly by a single 32-bit integer. Kennedy and Naaman [3] likewise apply the Shazam fingerprint [4] in a pairwise manner to synchronize videos of live concert recordings. The Shazam approach identifies the locations of spectral peaks in the spectrogram, considers various pairings of spectral peaks, and encodes the frequencies and time difference of each peak pair in a 32-bit fingerprint. Su et al. [5] extend the work of Kennedy and Naaman by applying a clustering technique to the pairwise match scores in order to group videos into sets of coherent scenes.

There is a rich literature on audio fingerprinting methods. Most prior work focuses on music identification. Some approaches use manually designed fingerprints based on spectral maxima locations [4][6], subband energy differences [2][7], wavelets [8], modulation frequency features [9] [10], spectral flatness [11] [12], and spectral subband moments [13]. Other approaches incorporate supervised learning into the fingerprint design process, such as selecting the best filters among a set of candidates through boosting [14][15] [16] or training a neural network [17]. In addition to music identification, the TRECVID content based copy detection task [18] also spurred research in online audio copy detection [7] [19] [20] [21].

Our current work explores an application of audio fingerprinting which has hitherto not been explored: aligning uncoordinated audio recordings of meetings, such as might be collected from participants’ cell phones. This application scenario presents some unique challenges and requirements which lead to novel developments of audio fingerprinting techniques. Our work has two main contributions. First, we propose an adaptive audio fingerprint which is learned on-the-fly on each alignment scenario (i.e., each set of audio files to align). Several of the above works incorporate supervised learning into the fingerprint design process, but, to the best of our knowledge, this is the first work which learns a fingerprint design in an unsupervised manner. This allows the fingerprint to adapt to the characteristics of each individual alignment scenario. Second, we propose a method for aligning a group of files which uses cumulative evidence. Rather than using an audio fingerprinting method out-of-the-box in a pairwise manner, we propose a way to align a group of files starting with the strongest matches first, and using the cumulative evidence of previously aligned files to help identify the weakest matches.

The paper is organized as follows. Section 2 describes the main components of the proposed alignment system. Section 3 explains the experimental setup. Section 4 shares the results and discussion. Section 5 concludes the work.

## 2. System Description

The alignment system will be described in three parts: the fingerprint computation, the spectrotemporal filter design, and the alignment algorithm.

### 2.1. Fingerprint Computation

Figure 2 shows a block diagram of the fingerprint computation process. There are five steps, each described below.

*Compute auditory spectrogram.* We computed a log mel spectrogram using 100 ms windows, 10 ms hop size, and 33 mel bands between 200Hz and 2kHz. These settings are similar to those used in previous audio fingerprinting works [2][14][8][7].

*Apply spectrotemporal filters.* We compute  $N$  features at each frame by applying  $N$  different spectrotemporal filters. In other words, each feature is a linear combination of the log mel spectrogram values for the current frame and surrounding context frames. Note that MFCCs are a special case of spectrotemporal filters in which the filter coefficients match the coefficients of the DCT transform. Rather than using MFCCs, however, we use filters that are learned in an unsupervised manner. We will discuss how to learn these filters in the next subsection.

*Compute deltas.* We compute deltas on the spectrotemporal features at a separation of  $T$  frames. If a feature at frame  $n$  is  $x_n$ , the delta feature will be  $\Delta x_n = x_n - x_{n+T}$ . The justification for this step will be explained in the next subsection.

*Compare threshold.* Each of the  $N$  delta spectrotemporal features is thresholded at 0, yielding a binary value.

*Bit packing.* The  $N$  binary values are packed into a single 32-bit integer which represents the fingerprint value for a single frame. This compact binary representation will allow us to store fingerprints in memory efficiently and to do reverse indexing to quickly look up fingerprint matches.

### 2.2. Fingerprint Design

We formulated the filter design as a series of optimization problems. Our formulation grows out of 3 design principles.

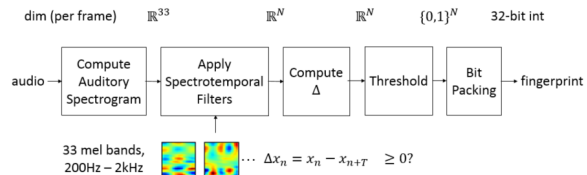


Figure 2: Block diagram of the fingerprint computation.

**Design principle 1: Compact.** A good fingerprint should represent information compactly. From an information theory perspective, the ideal 32-bit fingerprint should have 32 bits of entropy. Thus, each bit should be balanced (i.e. 0 half the time, and 1 half the time) and the bits should be uncorrelated. Any imbalance or correlation between bits represents inefficiency.

**Design principle 2: Robust.** A good fingerprint should be robust to noise. In the context of our fingerprint design where each bit represents a feature compared to a threshold, achieving robustness corresponds to maximizing the variance of the feature distribution. To see this, note that the feature distribution will be roughly bell-shaped (as a result of the central limit theorem), and that the threshold will be set at the median of the distribution (to achieve balanced bits). If a particular feature value falls close to the threshold, a small perturbation from noise may cause the feature to fall on the other side of the threshold, resulting in an incorrect bit. This situation can be minimized by maximizing the variance of the feature distribution.

**Design principle 3: Lightweight.** A good fingerprint should be lightweight to compute. We would like to avoid lots of dense multiplications. In the context of our filter design, we will restrict our attention to additions and subtractions only.

Putting these three principles together, we can formulate our fingerprint design in the following way. Consider the  $n^{th}$  audio frame in a set of training data, and let the log mel spectrogram values for the  $w$  context frames be denoted  $a_n \in \mathfrak{R}^{33w}$ . Let  $A \in \mathfrak{R}^{M \times 33w}$  denote the matrix containing all such data points  $a_n$ , where  $M$  is (approximately) the total number of audio frames in the training set. Let  $x_i \in \mathfrak{R}^{33w}$  specify the weights of the  $i^{th}$  spectrotemporal filter, and let  $S \in \mathfrak{R}^{33w \times 33w}$  be the covariance matrix of the data in  $A$ . Finally, let  $l$  denote the number of bits in the fingerprint. Then, for  $i = 1, 2, \dots, l$ , we would like to solve

$$\begin{aligned} & \text{maximize} && x_i^T S x_i \\ & \text{subject to} && x_i \in \{-1, 0, 1\}^{33w} \\ & && x_i^T x_j = 0, \quad j = 1, \dots, i - 1. \end{aligned}$$

Each resulting  $x_i$  specifies the spectrotemporal filter weights for the  $i^{th}$  fingerprint bit. The threshold for the  $i^{th}$  fingerprint bit is set to the median of the features  $Ax_i$ .

Let’s unpack the above formulation. The first line can be summarized as “maximize the variance.” To see this, note that the variance of the features  $Ax_i$  can be expressed as  $\frac{1}{M} \|\tilde{A}x_i\|_2^2$ , where the columns of  $\tilde{A}$  are zero mean. This objective is motivated by our second design principle (robust). The first constraint simply says, “addition and subtraction only.” The  $+1$  and  $-1$  correspond to addition and subtraction, and the  $0$  corresponds to simply ignoring an element. This constraint is motivated by our third design principle (lightweight). The last constraint says, “uncorrelated filters.” This constraint ensures that the filters are mutually orthogonal. This constraint is motivated by our first design principle (uncorrelated bits).

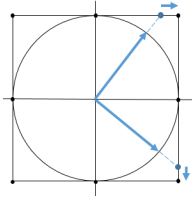


Figure 3: Projecting eigenvectors onto the  $\{-1, 0, 1\}^2$  lattice.

The above formulation is nice, but it is not tractable. The feasible set has  $3^{33w}$  possible values, which is too many to use a brute force method. Additionally, the objective requires *maximizing* a convex expression, which results in a nonconvex problem, so standard relaxation techniques cannot be applied naively. However, we can compute an approximate solution by relaxing the first constraint to  $\|x_i\|_2 = 1$ , and then projecting the solution back onto the  $\{-1, 0, 1\}^{33w}$  lattice. This approximation can be solved *exactly* using an eigenvalue decomposition. Now, for  $i = 1, \dots, l$ , we solve

$$\begin{aligned} & \text{maximize} && x_i^T S x_i \\ & \text{subject to} && \|x_i\|_2^2 = 1 \\ & && x_i^T x_j = 0, \quad j = 1, \dots, i-1. \end{aligned} \quad (1)$$

This is the eigenvalue problem, for which very efficient methods exist. We project the solution back onto the original feasible set by extending each eigenvector until it hits the surface of the unit hypercube, and then rounding each element to  $-1, 0$ , or  $1$ . Figure 3 shows a graphical depiction of this process in  $\mathbb{R}^2$ . In this case, the projected coordinates of the 2 eigenvectors remain orthogonal. In general, this will not hold for higher dimensions, though the projected coordinates will be largely uncorrelated (since they are roughly in the same direction as the eigenvectors). We simply accept these slight correlations as an acceptable cost that buys us a computationally efficient solution.

If we threshold the spectrotemporal features directly, the resulting fingerprint will not satisfy one very important characteristic: invariance to volume changes. This is important because when a person speaks, the same signal will be picked up by multiple recording nodes, but with varying attenuation levels depending on the distance to the speaker. To make our fingerprint invariant to volume, we compute deltas on the spectrotemporal features. Delta features will have a distribution centered around 0, so the median thresholds will all be set to 0. The fingerprint bits thus indicate whether each spectrotemporal feature is increasing or decreasing in time. This information is invariant to volume level as long as the speakers' locations are roughly stationary over short time intervals. Since spectrotemporal features for immediately adjacent frames will be highly correlated and thus yield delta features with very low variance, we compute deltas at a separation of 50 frames (.5 sec). This provides a reasonable tradeoff between minimizing correlation between features and ensuring the fingerprint is localized in time.

To recap, our fingerprint design is determined by: (1) determining the covariance matrix  $S$  of spectrogram values in  $w$  consecutive frames, (2) computing the top  $l$  eigenvectors  $x_1, \dots, x_l$ , (3) projecting each  $x_i$  onto  $\{-1, 0, 1\}^{33w}$  to get  $x_i^*$ , and (4) computing deltas on the features  $Ax_i^*$  at a separation of  $T$  frames. In addition to being compact, robust, and lightweight, the derived fingerprint design will be adapted to the data, invariant to changes in volume, and efficient to solve.

### 2.3. Alignment Algorithm

Now we describe the algorithm used to align a set of audio recordings. There are four steps, each described below.

**Step 1: Initialization.** The initialization step has 3 components. First, we learn the adaptive fingerprint design as described above. Second, we compute fingerprints on all the data and create a reverse index which maps each fingerprint value to the list of files and offsets at which the fingerprint occurs. Third, we select one channel to provide a universal time reference. All other time indices will be computed relative to the beginning of this ‘‘anchor’’ file.

**Step 2: Find the best match.** Using the anchor file as a query, we find the audio recording and time offset that has the strongest match. We compute a match score for every unaligned audio recording  $U_i$  by accumulating a histogram of offsets  $\Delta t$ , where  $\Delta t = \text{offset}_{\text{query}} - \text{offset}_{U_i}$  is the relative offset between two matching fingerprints. There will be many matching fingerprints at the true relative offset, so we take the maximum value of the histogram counts as the match score. This approach is adopted from [4].

**Step 3: Fine alignment.** We consider a range of possible offsets around the best  $\Delta t$  and compute a more fine-grained match score for  $U^*$ , the unaligned audio recording with the best (rough) match score. For each possible offset, we determine the fingerprint bit agreement rate between  $U^*$  and all of the currently aligned files. These bit comparisons can be computed very efficiently using bit arithmetic. The offset  $\Delta t^*$  with highest bit agreement is the final alignment estimate for  $U^*$ .

**Step 4: Repeat steps 2 and 3.** We repeat step 2 using the most recently aligned file as the query file. For all aligned files, frame offsets are adjusted to represent the universal time index. At each stage, we retain the histograms from previous steps and simply add additional counts. In this way, we accumulate more and more evidence to help align the weakest matches.

## 3. Experimental Setup

We ran experiments on data extracted from the ICSI meeting corpus [22]. The original data set consists of multi-channel audio recordings of 75 research group meetings, totaling approximately 72 hours of meetings. The multi-channel recordings include close-talking microphones and 6 omnidirectional tabletop microphones of varying quality. Typical meetings are about an hour long, and the number of simultaneous channels ranged from 9 to 15. The corpus is a suitable data set because it has diversity in both microphone location and microphone characteristics, as would be the case if participants used their cell phones as recording devices.

Given a set of meeting recordings, we can generate an alignment scenario by extracting a random segment from each channel. The length of these segments is selected randomly according to a  $[0, 600\text{sec}]$  uniform distribution, and we ensure that each segment overlaps with at least one other segment by 30 seconds or more. Since the above process is probabilistic, we can generate multiple alignment scenarios from a single meeting. We generated 10 query scenarios from each of the 75 meetings, resulting in a total of 750 alignment scenarios and approximately 8500 alignments. We used 37 meetings for debugging and determining appropriate system hyper-parameters (note that we do not require training, since our method works in an unsupervised manner), and we used the other 38 meetings for testing. Note that the queries we generated are probably more difficult and challenging than a typical use case scenario, since users would

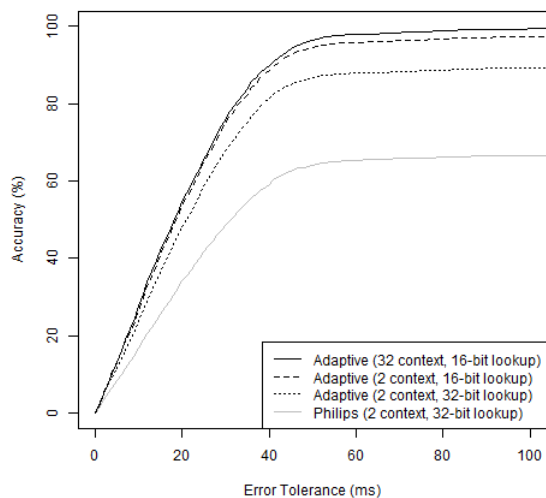


Figure 4: The tradeoff between accuracy and error tolerance for the alignment system with four different fingerprints.

all tend to record a very substantial chunk of the meeting, with an occasional user leaving the meeting early or entering very late. However, generating more difficult query scenarios like this will enable us to better characterize and test the robustness of our system.

To evaluate our system, we compare each estimated alignment with the true alignment. Let  $e$  denote the difference between the estimated and true alignment (e.g. in figure 1,  $e_B = \Delta B_{hyp} - \Delta B_{ref}$ ). If  $|e| > \gamma$ , we consider that particular alignment to be incorrect. By considering a range of  $\gamma$  values, we can characterize the tradeoff between accuracy and error tolerance. Our accuracy versus error tolerance curves aggregate the results over all alignment scenarios.

## 4. Results and Discussion

Figure 4 shows the tradeoff between accuracy and error tolerance for the alignment system with 4 different fingerprints:

1. Philips (2 context frames, 32-bit lookup)
2. Adaptive (2 context frames, 32-bit lookup)
3. Adaptive (2 context frames, 16-bit lookup)
4. Adaptive (32 context frames, 16-bit lookup)

We selected these four systems to tease out the effect of different factors on the overall performance. The lowest curve shows the performance of the Philips fingerprint, which serves as a baseline comparison. The Philips fingerprint was described previously, and is the most highly cited work in the literature [2]. The gap between the first and second curves shows the benefit of switching from the Philips design to the adaptive design, when both are given the same amount of context frames and fingerprint bits. The gap between the second and third curves shows the benefit of reducing the number of lookup bits in the adaptive fingerprint. The gap between the third and fourth curves shows the benefit of increasing the amount of context in the adaptive fingerprint.

We can see that there is substantial improvement in switching from the Philips design to the adaptive design, improving

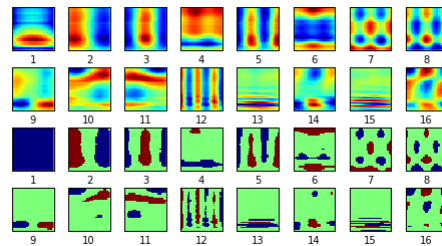


Figure 5: The top 16 learned filters from one alignment scenario before projection (top) and after projection (bottom).

the accuracy at 100 ms tolerance from 66.5% to 89.3%. There is also substantial improvement when reducing the number of fingerprint bits from 32 to 16, boosting the accuracy at 100 ms tolerance from 89.3% to 97.2%. (For a discussion of the optimal number of bits, see [23].) There is an additional improvement to 99.4% when increasing the amount of context from 2 to 32 frames, though at this point the results are nearly saturated, so it is difficult to determine exactly how much robustness is gained. With all of these improvements combined, the adaptive fingerprints improve the accuracy from 66.5% to 99.4%.

Another question of interest is to examine what the learned filters look like. Figure 5 shows the top 16 learned filters with 32 frames of context for one example alignment scenario. The top two rows show the filters before projection, and the bottom two rows show the filters after projection. The filters are arranged first from left to right, and then from top to bottom.

There are three things to notice about the filters in figure 5. First, the filters capture modulations in both time and frequency. Some filters capture modulations in the temporal dimension (e.g. 2,3,5,12), some in the spectral dimension (e.g. 4,6,13,15), and others in both dimensions (e.g. 7,8,11,16). The important thing to notice is that both types of modulations are important. Thus, only considering 2 frames of context would significantly hinder the fingerprint’s representational power, since 2 frames would be insufficient to capture variations in time. Second, low modulation frequencies seem to be most important. We see a progression from slow modulations to fast modulations as we progress to later and later filters. For example, filters 2, 3, 5, and 12 capture faster and faster modulations in time. Third, many of the projected filter coefficients are 0. So, in addition to only requiring additions and subtractions, these filters have the added benefit of being able to ignore many elements.

## 5. Conclusion

We have proposed an adaptive audio fingerprint which can be learned on-the-fly to robustly and efficiently align a set of temporally overlapping meeting recordings. We have also introduced an algorithm for aligning sets of files which uses the cumulative evidence of previous alignments to help align the weakest matches. On a set of alignment scenarios extracted from the ICSI meeting corpus, the proposed method demonstrates significant improvement over the well-known Philips fingerprint and is able to achieve  $> 99\%$  alignment accuracy.

## 6. Acknowledgements

Thanks to Adam Janin, Nelson Morgan, Steven Wegmann, and Eric Chu for constructive feedback and discussions.

## 7. References

- [1] P. Shrestha, M. Barbieri, and H. Weda, "Synchronization of multi-camera video recordings based on audio," in *Proc. ACM International Conference on Multimedia*, 2007, pp. 545–548.
- [2] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. International Society for Music Information Retrieval (ISMIR'02)*, Paris, France, Oct. 2002, pp. 107–115.
- [3] L. Kennedy and M. Naaman, "Less talk, more rock: Automated organization of community-contributed collections of concert videos," in *Proc. ACM International Conference on World Wide Web*, 2009, pp. 311–320.
- [4] A. L.-C. Wang, "An industrial-strength audio search algorithm," in *Proc. International Society for Music Information Retrieval (ISMIR'03)*, Baltimore, Maryland, USA, Oct. 2003, pp. 7–13.
- [5] K. Su, M. Naaman, A. Gurjar, M. Patel, and D. P. W. Ellis, "Making a scene: Alignment of complete sets of clips based on pairwise audio match," in *Proc. ACM International Conference on Multimedia Retrieval (ICMR'12)*, 2012.
- [6] S. Fenet, G. Richard, and Y. Grenier, "A scalable audio fingerprint method with robustness to pitch-shifting," in *Proc. International Society for Music Information Retrieval (ISMIR'11)*, 2011, pp. 121–126.
- [7] X. Anguera, A. Garzon, and T. Adamek, "MASK: Robust local features for audio fingerprinting," in *Proc. IEEE International Conference on Multimedia and Expo (ICME'12)*, Melbourne, Australia, Jul. 2012, pp. 455–460.
- [8] S. Baluja and M. Covell, "Waveprint: Efficient wavelet-based audio fingerprinting," *Pattern Recognition*, vol. 41, no. 11, pp. 3467–3480, May 2008.
- [9] S. Sukittanon and L. E. Atlas, "Modulation frequency features for audio fingerprinting," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, 2002, pp. 1773–1776.
- [10] M. Ramona and G. Peeters, "Audio identification based on spectral modeling of bark-bands energy and synchronization through onset detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'11)*, 2011, pp. 477–480.
- [11] J. Herre, E. Allamanche, and O. Hellmuth, "Robust matching of audio signals using spectral flatness features," in *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, New Platz, New York, USA, Oct. 2001, pp. 127–130.
- [12] E. Allamanche, J. Herre, O. Hellmuth, B. Fröba, T. Kastner, and M. Cremer, "Content-based identification of audio material using MPEG-7 low level description," in *Proc. International Society for Music Information Retrieval (ISMIR'01)*, 2001.
- [13] J. S. Seo, M. Jin, S. Lee, D. Jang, S. Lee, and C. D. Yoo, "Audio fingerprinting based on normalized spectral subband moments," *IEEE Signal Processing Letters*, vol. 13, no. 4, pp. 209–212, 2006.
- [14] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, California, USA, Jun. 2005, pp. 597–604.
- [15] D. Jang, C. D. Yoo, S. Lee, S. Kim, and T. Kalker, "Pairwise boosted audio fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 4, pp. 995–1004, 2009.
- [16] S. Kim and C. D. Yoo, "Boosted binary audio fingerprint based on spectral subband moments," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'07)*, Honolulu, Hawaii, USA, Apr. 2007, pp. 241–244.
- [17] C. J. C. Burges, J. C. Platt, and S. Jana, "Distortion discriminant analysis for audio fingerprinting," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 3, pp. 165–174, 2003.
- [18] P. Over, G. Awad, J. Fiscus, B. Antonishek, M. Michel, A. F. Smeaton, W. Kraaij, and G. Quénot, "TRECVID 2011 - An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics," in *TRECVID 2011 - TREC Video Retrieval Evaluation Online*, Gaithersburg, Maryland, USA, Dec. 2011.
- [19] V. N. Gupta, G. Boulianne, and P. Cardinal, "CRIM's content-based audio copy detection system for TRECVID 2009," *Multimedia Tools and Applications*, vol. 60, no. 2, pp. 371–387, 2012.
- [20] E. Younessian, X. Anguera, T. Adamek, N. Oliver, and D. Marimon, "Telefonica research at TRECVID 2010 content-based copy detection," in *Proc. of TRECVID 2010*, 2010.
- [21] Y. Uchida, S. Sakazawa, M. Agrawal, and M. Akbacak, "KDDI labs and SRI international at TRECVID 2010: Content-based copy detection," in *Proc. of TRECVID 2010*, 2010.
- [22] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI Meeting Corpus," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, 2003, pp. 364–367.
- [23] T. Tsai, G. Friedland, and X. Anguera, "An information-theoretic metric of fingerprint effectiveness," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'15)*, 2015, to appear.