SURVEY AND EVALUATION OF AUDIO FINGERPRINTING SCHEMES FOR MOBILE QUERY-BY-EXAMPLE APPLICATIONS

Vijay Chandrasekhar vijayc@stanford.edu
Matt Sharifi mns@google.com
David A. Ross dross@google.com

ABSTRACT

We survey and evaluate popular audio fingerprinting schemes in a common framework with short query probes captured from cell phones. We report and discuss results important for mobile applications: Receiver Operating Characteristic (ROC) performance, size of fingerprints generated compared to size of audio probe, and transmission delay if the fingerprint data were to be transmitted over a wireless link. We hope that the evaluation in this work will guide work towards reducing latency in practical mobile audio retrieval applications.

1. INTRODUCTION

Audio fingerprinting provides the ability to derive a compact representation which can be efficiently matched against other audio clips. With smart phones becoming ubiquitous, there are several applications of audio fingerprinting on mobile devices. A common use case is query-by-example music recognition: a user listens to a song in a restaurant, shopping mall, or in a car, and wants to know more information about the song. Shazam [1] and SoundHound [2] are examples of popular music recognition applications on cellphones. Other applications of audio fingerprinting on mobile devices include copyright detection [4], personalized entertainment and interactive television without extraneous hardware [8].

Mobile query-by-example applications pose a unique set of challenges. First, the application has to be low-latency to provide users with an interactive experience. To achieve low latency, the retrieval framework has to adapt to stringent memory, computational, power and bandwidth requirements of the mobile client. It is important that the size of the data generated needs to be as small as possible to reduce network latency, which is typically the bottleneck in 3G networks. Second, the length of the audio required to get a match should be short for mobile applications (e.g., $<$10 seconds). Current applications Shazam [1] and SoundHound [2] often require $>$10 seconds for retrieval. For copyright detection, one might use 30-60 second probes for retrieval [4], which is not feasible for interactive mobile applications. Third, the distortions introduced by cell phones tend to be more severe than simple degradations like compression artifacts, time-offsets, amplitude compression or structured noise present in near-duplicate detection problems [4]. On mobile devices, we need to be mindful of ambient noise present in shopping malls or cafes, errors in sampling through telephony equipment, low bit-rate voice compression and other quality-enhancement algorithms that might be built into the mobile device or introduced by the carrier network. In this work, we evaluate the state-of-the-art in content-based audio retrieval with focus on query-by-example mobile applications.

2. PRIOR WORK AND MOTIVATION

State-of-the-art audio retrieval applications use a set of low level fingerprints extracted from the audio sample for retrieval. The fingerprints are typically computed on the spectrogram - a time frequency representation of the audio. Haitsma et al. [11] propose fingerprints based on Bark Frequency Cepstrum Coefficients (BFCC). Highly overlapping frames are considered to ensure that the query probe can be detected at arbitrary time-alignment. Each fingerprint is 32 bits and can be compared efficiently with Hamming distances. Ke et al. [14] improve the performance of the fingerprinting scheme in [11] using the AdaBoost technique from computer vision. Baluja et al. [4] propose a scheme based on wavelets. The overlapping spectrogram images are transformed into a sparse wavelet representation and the popular min-hash technique [5] is used to obtain a 100 byte fingerprint which can be compared directly with byte-wise Hamming distances. In contrast to the three schemes above, Wang [17, 18] proposes looking only at spectrogram peaks.

The authors are not aware of a comprehensive evaluation of the different fingerprinting schemes in a common framework. In contrast, several such evaluations exist for image features in the computer vision community for content-based image retrieval [15, 19]. Fingerprints developed for

applications like query-by-humming and cover song detection are outside the scope of this paper. In particular, we are interested in factors affecting practical query-by-exact-example mobile applications. The questions that are most critical for mobile applications are:

- How much fingerprint data does each scheme generate?
- How does the size of the fingerprint data compare to the size of the compressed audio needed for accurate retrieval?
- What would the transmission delay be if the fingerprints were transmitted over a typical 3G network?
- How discriminative are the different fingerprinting schemes?
- How do the different schemes perform for really short($\sim$5 seconds) and noisy query probes captured by cell phones ?
- How does the performance of each scheme vary as a function of probe length in the range of 5 to 15 seconds typical for mobile applications?

3. CONTRIBUTIONS

We survey and evaluate popular audio fingerprinting schemes in a common framework with short noisy query audio probes captured from cell phones. We report and discuss results important for mobile applications: Receiver Operating Characteristic (ROC) performance, size of fingerprints generated compared to size of audio probe, and transmission delay if the fingerprint data were to be transmitted over a wireless link. We hope that the evaluation in this paper will provide key insights and guide us towards developing low latency retrieval systems. In Section 4, we survey the different audio fingerprinting schemes. In Section 5, we describe the evaluation framework and provide experimental results.

4. SURVEY OF FINGERPRINTING SCHEMES

Before we survey popular audio fingerprinting schemes, we discuss the typical pipeline for audio retrieval applications. First, a set of fingerprints are extracted from the query song. The fingerprints could be extracted at uniform sampling rate, or only around points of interest in the spectrogram (e.g., spectrogram peaks in the case of Wang [18]). For mobile applications, it is critical that individual fingerprints be robust against ambient noise, compared to the corresponding database fingerprint.

Next the query is compared with a database of reference tracks to find candidate matches. To avoid pairwise comparison between the query and all of the reference tracks, the database is partitioned. The partitioning of the database is precomputed for the database, and each partition is associated with a list of database songs (also called an inverted index). The partitioning on the database could be done by direct hashing of the fingerprints (e.g., a 32 bit fingerprint could be directly hashed into a table with 4 billion entries), Locality Sensitive Hashing or techniques based on Vector Quantization. This partitioning allows approximate-nearest-neighbor-search as exact-nearest-neighbor search is infeasible in a database with billions of fingerprints. The inverted file for each cell consists of a list of song IDs and the timing offsets at which the fingerprints appear. The timing information is used in the final step of the pipeline. Based on the number of fingerprints they have in common with the query probe from the inverted index, a short list of potentially similar database songs is selected from the database.

Finally, a temporal alignment step is applied to the most similar matches in the database. Techniques like Expectation Maximization [14], RANSAC [9], or Dynamic Time Warping [6] are used for temporal alignment. In the case of linear correspondence (i.e., the tempo of the database and query songs are the same), Wang [18] proposes using a simple and fast technique that looks for a diagonal in the time-vs-time plot for matching database and query fingerprints. The existence of a strong diagonal indicates a valid match. The temporal alignment step is used to get rid of false positives, and enables very high precision retrieval.

In this Section, we review three fingerprinting schemes in detail: Ke [14], Baluja [4] and Wang [18]. In the interest of space, we omit the scheme proposed by Haitsma [11] as the fingerprint by Ke improves directly upon their scheme [14]. For a comparison of the two schemes by Ke and Haitsma, interested readers are referred to [14]. For each scheme, we first discuss the details of the scheme and the motivation behind the approach, followed by system parameters suggested by the authors that provide good trade-off between accuracy and computational complexity.

4.1 Ke, Hoiem and Sukthankar

4.1.1 Description

Ke's approach builds on popular classification techniques in the computer vision community. Ke provides the important insight that 1-D audio signals can be processed as conventional images when viewed in the time-frequency spectrogram representation. The time-frequency spectrogram data is treated as a set of overlapping images. To compute a compact fingerprint on each image, the authors first train simple AdaBoost classifiers based on box-filters, a technique popular in face detection. The training data for classification is obtained by considering audio samples and their corresponding versions degraded by noise. The output of each classifier yields a binary value. E.g., each classifier outputs a 1 or a 0 based on the differences between values aggregated in two sub-rectangular regions of the spectrogram image. The concatenated output of the set of classifiers is then used as a fingerprint of the spectrogram image.

4.1.2 System Parameters

Ke and Haitsma use the same set of parameters for computing the spectrogram. The spectrogram, obtained by Short Term Fourier Transform (STFT), represents the power in 33 logarithmically-spaced frequency bands spaced 300 Hz and 2000 Hz. Overlapping spectrogram images measured over 0.372s windows are considered in 11.6 ms increments ($\sim$100 fingerprints/second). The short increments coupled with large spectrogram images at each step are used to make the scheme robust to sampling errors and small time-offsets. For a 10 second probe, the scheme produces 860 fingerprints. For the AdaBoosting step, 32 classifiers are chosen out of a candidate list of 25000 filters. We use the training data sets and code provided by the authors at [13]. Two fingerprints are considered to be a match if they have a Hamming distance $<$2, in the feature matching step of the retrieval pipeline.

4.2 Baluja and Covell

4.2.1 Description

Similar to Ke's work, Baluja's fingerprint is also inspired from the image retrieval community. The pipeline for computing "waveprints"(the term used by the authors to describe their wavelet-based fingerprints) is illustrated in Fig. 1, and in inspired from [12].

First, the authors compute overlapping spectrogram images using the same approach proposed by Ke. Next, the spectrogram images are decomposed using multi-resolution Haar wavelets. Wavelets are chosen due to their effectiveness in the retrieval work presented in [12]. An image produces as many wavelet co-efficients as pixels. Next, the authors retain only the top-t few wavelets, where t is chosen to be much smaller than the size of the spectrogram image. Next, the authors observe that the top-t wavelets are sparse. To obtain a compact represenation, the authors only retain the sign information (an approach also found effective in [12]), and use the Min-Hash technique to generate a set of p bytes that is used to represent the original spectrogram image. Two spectrogram images can now be compared directly by computing the byte-wise Hamming distance of the p bytes. For this approach to be effective, p needs to be large (typically chosen to be 100). Nearest neighbor searching in a 100 dimensional space is non-trivial. Hence, in the final step, Locality Sensitive Hashing (LSH) is used to find approximate-nearest-neighbor fingerprints in this space.

4.2.2 System Parameters

The authors optimize system parameters for accuracy and computational complexity in [3, 4]. We use the parameters recommended by the authors in [3]. Overlapping spectrogram images measured over 0.372 second windows are considered in 0.09 second strides ($\sim$10 fingerprints/second). $t = 200$ top wavelets are considered. p is chosen to be 100, i.e., each fingerprint is represented as 100 bytes. For LSH, the 100-byte fingerprint is divided into 25 equal 4-byte bands. Each 4-byte band is stored as a 32 bit hash table. In the feature-matching step, two fingerprints are considered to be a match if their 4-byte representations match in at least one of the 25 LSH bands.

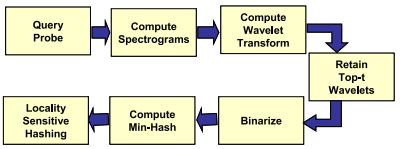


Figure 1. Pipeline for extracting waveprint features proposed by Baluja [4]. Spectrogram images are represented as p bytes obtained from Min-Hashing, which can be compared byte-wise directly for computing similarity.

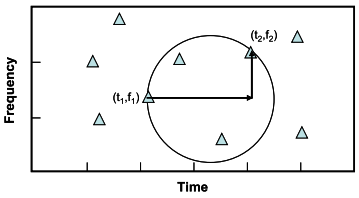


Figure 2. Illustration of audio fingerprints proposed by Wang [17]. Triplet information $((t_2 - t_1, f_1, (f_2 - f_1))$ is quantized to form the fingerprint.)

4.3 Wang

4.3.1 Description

While the schemes by Ke and Baluja use dense sampling and compute fingerprints over fairly large spectrogram images, Wang proposes looking only at spectrogram peaks. There are two reasons for choosing spectrogram peaks: First, spectrogram peaks are more likely to survive ambient noise. Second, spectrogram peaks satisfy the property of linear superposition, i.e., a spectrogram peak analysis of music and noise together will contain spectral peaks due to the music and the noise as if they were analyzed separately [17]. The fingerprinting scheme is illustrated in Fig. 2. For pairs of peaks (t_1, f_1) and (t_2, f_2), the fingerprint is computed on a triplet of $((t_2 - t_1), f_1, (f_2 - f_1))$. Each number in the triplet is quantized and the concatenated value is treated as the fingerprint.

4.3.2 System Parameters

For this scheme, we adapt the implementation provided by Ellis [7]. We optimize over a parametric space, and choose the following set of parameters. The frequency data in the spectrogram is divided into 256 levels linearly. We consider neighboring peaks in an adjacent frequency range of 64 units, and timing range of 64 units (sampling rate of the audio signal is set to 8 KHz). The values $((t_2-t_1), f_1, (f_2-$

f_1)) are represented as 6,8 and 6 bits respectively to obtain a 20 bit fingerprint. For this data set, the 20 bit fingerprint works better than a 32-bit fingerprint suggested by Wang in [18] - note that over quantization could affect performance adversely. We generate 20 fingerprints per second.

5. EXPERIMENTAL RESULTS

We use our own data set as we were not able to find any publicly available data sets captured from mobile phones. Most existing data sets introduce artificial distortions to the audio (e.g., adding noise), and are not representative of distortions typical in the mobile scenario. We captured audio clips on a Nexus One phone from a set of 39 songs played on TV and from laptop speakers in noisy environments. In our data collection, we tried to capture noise from different ambient noise sources. Our song data set contains popular songs from artists like Lady Gaga, Michael Jackson, Green Day, Avril Lavigne, to name a few. Each of these clips is between 60 and 90 seconds long, which we divide into non-overlapping 5, 10 and 15 second snippets to use as query probes. This gives us a ground truth data set of over a 1000 pairs of query probes and their corresponding uncorrupted reference songs. All pairs between query and reference, both positive and negative examples, are considered to generate Receiver Operating Characteristic (ROC) curves.

5.1 Receiver Operating Characteristic

We evaluate the different fingerprinting schemes first after the fingerprint indexing step, and subsequently, the temporal alignment step.

5.1.1 Fingerprint Indexing

The inverted index on the database enables fast retrieval and provides a shortlist of candidates to be considered for a more extensive temporal alignment check. Each query fingerprint votes for all the database fingerprints that it finds in the inverted index. The similarity between the database song and query song is the number of fingerprints in common between them, based on the approximate-nearest-neighbor indexing strategy. For Ke, the similarity measure is the number of fingerprints that have <2 Hamming distance. For Baluja, the similarity measure is the number of fingerprints that have $>=1$ matches in the 25 LSH bands. For Wang, the similarity measure is the number of 20-bit fingerprints that get hashed to the same bin.

We compute such a similarity score for matching and non-matching pairs of ground-truth query and database songs, for the different schemes. From these similarity scores, we form two histograms, one for matching pairs and one for non-matching pairs, as illustrated in Fig. 3. The overlapping between the two histograms depends on the fingerprinting scheme, and more importantly, the length of the query probe. The longer the query probe, the lower the overlap between the two histograms, and the better the performance of the scheme. Also, the more discriminative the fingerprint, the lower the overlap between the two histograms. From the two histograms we obtain a Receiver Operating Characteristic (ROC) curve which plots correct match fraction against incorrect match fraction. The different points on the ROC curve are obtained by adjusting the similarity measure threshold. The higher the ROC curve, the more effective the retrieval system.

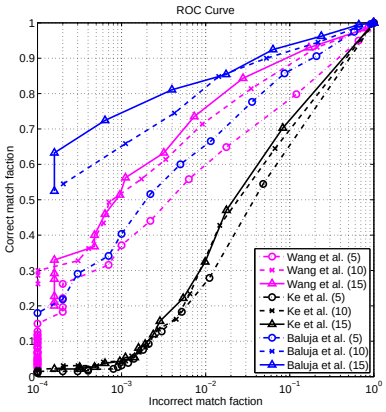


Figure 4. ROC performance of different schemes. The number in brackets is the length of the query probe in seconds. The performance of each fingerprinting scheme increases as the query length increases. Baluja's scheme performs the best.

We plot the ROC performance of the three schemes in Fig. 4. For each scheme, we note that the ROC performance improves as the length of the query probe increases from 5 to 15 seconds, as expected. Typically, the returns are diminishing beyond 10 seconds. Baluja's fingerprinting scheme performs the best for all query probe lengths. The MinHash based fingerprints (100 bytes each) are highly discriminative and capture information over a longer time-duration than Wang's scheme.

The Wang fingerprints are far more compact - however, the fingerprints are sensitive to small offsets in spectrogram peak localization. The low dimensionality of the fingerprint makes it less discriminative, causing the scheme to require a longer probe to achieve a comparable performance to Baluja's scheme. Also, the lower dimensionality of the descriptor implies that it does not scale well as the size of the database grows. As the length of the query probe increases to 15 seconds, Wang's scheme catches up in performance.

Finally, we observe that Ke's scheme performs poorly for the short query probes that we are interested in. For Ke's scheme to catch up in ROC performance, much longer probes would be required. The scheme also suffers due to its dependence on the set of AdaBoost classifiers used to generate the fingerprint. For our evaluation, we used the AdaBoost classifiers provided by the authors in [13]. A

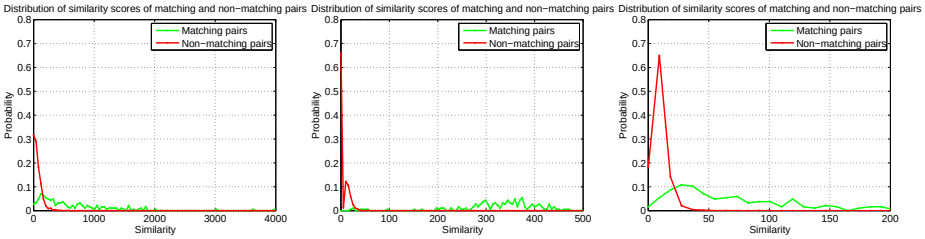


Figure 3. Distribution of scores for matching and non-matching pairs of query probe and reference songs illustrated for the different fingerprinting schemes. Ideally, we would like to have the matching pairs to have very high scores, and non-matching pairs to be exactly 0. The overlap in the distributions causes errors in retrieval. This overlap depends on the discriminativeness of the fingerprinting scheme and also on the length of the query probe. Longer query probes provide a better separation between the two distributions.

mismatch between training and test data can affect the performance of this scheme adversely. We require robustness against a broad range of mobile environments and noise sources, and training a set of AdaBoost classifiers for different environments is not practical.

5.1.2 Temporal Alignment

Based on computational resources available, accuracy requirements and the size of the database, retrieval systems choose an operating point on the curve shown in Fig. 4. E.g., state-of-the-art retrieval systems would typically operate in the 80-90% True Positive Rate regime. At the operating point, we apply the Temporal Alignment (TA) scheme proposed by Wang to get rid of false positives. It is relatively easy to achieve high precision for audio retrieval applications. By requiring a minimum number of fingerprint matches to satisfy TA, we can get rid of most false positives. We set the minimum number of temporally aligned matches to 5 for this experiment. We plot the percentage of queries passing the temporal alignment check as a function of query probe length in Fig. 5. Again, we observe Baluja's scheme performs the best, followed by Wang and Ke respectively. The performance for each scheme improves as the length of the query probe increases. We conclude that highly discriminative fingerprints help significantly for short 5 second query probes. Next, we study the amount of data generated for each fingerprinting scheme.

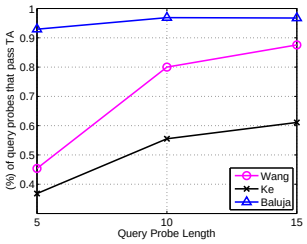


Figure 5. Recall as a function of query probe length for different schemes. Precision is 100% as the temporal alignment step eliminates false positives.

5.2 Data Size and Transmission Delay

The different fingerprinting schemes generate different amounts of data. Here, we present results for a 10 second probe, as 10 second probes provide a balance between accuracy and latency for all three schemes. Ke's scheme produces 729 4-byte fingerprints, Baluja's scheme produces 87 100-byte fingerprints, and Wang's scheme produces 587 20-bit fingerprints on average for 10 second probes. The amount of data generated for the different schemes is shown in Fig. 7. We compare the size of fingerprint data to the size of a 10 second Vorbis compressed audio at 64 kbps (80 KB). We observe that the size of fingerprint data is significantly lower than the size of the compressed audio for all fingerprinting schemes (<10 KB). This motivates computing the fingerprints on the device, whenever possible. We note that Wang's scheme produces less data than Baluja's or Ke's scheme. For a fair comparison between the different schemes, we plot the bitrate-Equal Error Rate (EER) performance in Figure 6. We note that the reduction in data for Wang's scheme comes at the cost of ROC performance shown in Fig. 6.

If fingerprinting were to be done on the device, how long would the transmission delay be for sending the fingerprint data? The transmission delay would depend on the wireless network used: 3G or WLAN (Wireless LAN). WLAN systems provide much higher bandwidth compared to 3G, and transmission delay is negligible even for large packet sizes. Here, we present transmission delay numbers only for a 3G connection, as it is the most prevalent on mobile phones today [16]. For network transmission delay experiments, we use the data presented in [10, 16]. The authors conduct experiments in an AT&T 3G wireless network, with a total of more than 5000 transmissions at locations where a typical audio retrieval system would be used.

We present the time it would take to transmit fingerprint data for the different schemes in Fig. 7(b). Transmitting fingerprint data takes in the order of a few seconds, while transmitting the compressed audio could take tens of seconds, based on the wireless link. Note that the delay numbers shown here only represent the data transmission delay for

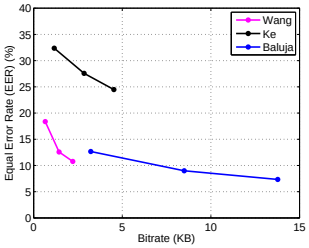
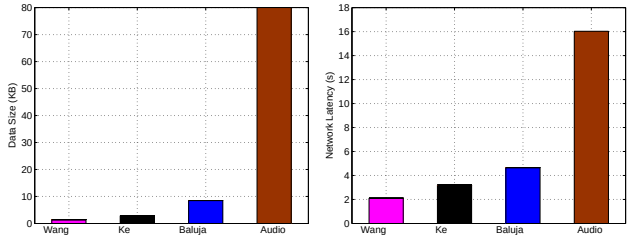


Figure 6. Equal Error Rate (EER) vs. bitrate tradeoff. Baluja scheme works well at high bitrates, while Wang's scheme works well at low bitrates.

different fingerprinting schemes. The end-to-end system latency would depend on the streaming protocol, the length of query probe considered, transmission delay and processing delay on the server. Based on the experimental results presented here and in [10], we would expect the transmission delay to be the bottleneck in 3G networks, which motivates computing fingerprints on the device.



(a) (b)

Figure 7. Fig.(a) shows size of data generated by different schemes. Fig.(b) shows the associated transmission delay if the data were to be transferred over a 3G network. The data and transmission delay numbers are presented for 10 second query probes. Data for 5 and 15 second probes can be extrapolated linearly.

Finally, we draw some parallels between mobile image retrieval and audio retrieval. We note that Ke and Baluja were both inspired by work in computer vision literature. Interest point detectors and descriptors have been well studied in computer literature: readers are referred to the survey papers [15, 19]. What has pushed the field forward is the availability of good image and patch level data sets that capture the distortions (e.g., perspective and lighting in images) that interest point detectors and descriptors need to be robust against. The availability of similar ground-truth data sets will be useful for designing interest point detectors and descriptors for audio retrieval. Spectrogram peaks proposed by Wang is one example of interest point detection, but other schemes need to be explored. Interest point detectors are the first step in the pipeline, and improvements here could affect blocks further down the pipeline. Next, we note that the best descriptors in the vision literature are high-dimensional and capture salient characteristics in a local neighborhood around the interest point. In the case of audio retrieval, we need descriptors around interest points to be robust against small timing offset errors, and distortions introduced by ambient noise. Both interest point detectors and descriptors for audio retrieval in highly noisy environments are interesting areas for future work. We conclude by noting that techniques and algorithms developed in recent image retrieval literature can be used to further improve efficiency and performance of audio retrieval systems.

6. CONCLUSION

We perform a thorough survey and evaluation of popular audio fingerprinting schemes in a common framework. We report and discuss results important for mobile applications: Receiver Operating Characteristic (ROC) performance, size of fingerprints generated compared to size of the compressed audio sample, transmission delay if the fingerprint data were to be transmitted over a 3G wireless link and computational cost of fingerprint generation.

7. REFERENCES

[1] *Shazam Music Recognition Service*. `http://www.shazam.com/`.

[2] *SoundHound*. `http://www.soundhound.com/`.

[3] S. Baluja and M. Covell. Audio fingerprinting: Combining computer vision and data stream processing. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA, April,2007.

[4] S. Baluja and M. Covell. Content fingerprinting using wavelets. In *Proc. of European Conference on Visual Media Production (CVMP)*, London, UK, Nov,2006.

[5] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang. Finding interesting associations without support pruning. In *Proc. of the 16th International Conference on Data Engineering (ICDE)*, 1999.

[6] M. Covell and S. Baluja. Known audio detection using waveprint: Spectrogram fingerprinting by wavelet hashing. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA, April,2007.

[7] D. Ellis. *Robust Landmark-Based Audio Fingerprinting*. `http://labrosa.ee.columbia.edu/matlab/fingerprint/`.

[8] M. Fink, M. Covell, and S. Baluja. Social and interactive-television applications based on realtime ambient-audio identificatio. In *Proc. of European Conference on Interactive TV (EuroITV)*, Athens, Greece, 2006.

[9] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 24(6):381–395, 1981.

[10] B. Girod, V. Chandrasekhar, D. M. Chen, N. M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham. Mobile Visual Search. In *Proceedings of IEEE Signal Processing Magazine, Special Issue on Mobile Media Search*, 2010.

[11] J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *Proc. of International Conference on Music Information Retrieval (ISMIR)*, Paris, France, 2002.

[12] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *Proc. of the 22nd annual Conference on Computer graphics and Interactive Techniques (SIGGRAPH)*, pages 277–286, New York, NY, USA, 1995. ACM.

[13] Y. Ke, D. Hoiem, and R. Sukthankar. *Software*. `http://www.cs.cmu.edu/~yke/musicretrieval/`.

[14] Y. Ke, D. Hoiem, and R. Sukthankar. Computer vision for music identification. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, USA, June,2005.

[15] K. Mikolajczyk and C. Schmid. Performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[16] S. S. Tsai, D. M. Chen, V. Chandrasekhar, G. Takacs, N. M. Cheung, R. Vedantham, R. Grzeszczuk, and B. Girod. Mobile Product Recognition. In *Proc. of ACM Multimedia (ACM MM)*, Florence, Italy, October 2010.

[17] A. Wang. The shazam music recognition service. *Communications of the ACM*, 49(8):44–48, 2006.

[18] A. Wang. An industrial-strength audio search algorithm. In *Proc. of International Conference on Music Information Retrieval (ISMIR)*, Baltimore, Maryland, USA, October,2003.

[19] S. Winder and M. Brown. Learning Local Image Descriptors. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Minneapolis, Minnesota, 2007.