AN INFORMATION-THEORETIC METRIC OF FINGERPRINT EFFECTIVENESS

TJ Tsai[†‡] *Gerald Friedland*[†‡] *Xavier Anguera*[⋆]

[†] University of California Berkeley, Berkeley, CA
[‡]International Computer Science Institute, Berkeley, CA
[⋆]Telefonica Research, Barcelona, Spain

ABSTRACT

Audio fingerprinting refers to the process of extracting a robust, compact representation of audio which can be used to uniquely identify an audio segment. Works in the audio fingerprinting literature generally report results using system-level metrics. Because these systems are usually very complex, the overall system-level performance depends on many different factors. So, while these metrics are useful in understanding how well the entire system performs, they are not very useful in knowing how good or bad the fingerprint design is. In this work, we propose a metric of fingerprint effectiveness that decouples the effect of other system components such as the search mechanism or the nature of the database. The metric is simple, easy to compute, and has a clear interpretation from an information theory perspective. We demonstrate that the metric correlates directly with system-level metrics in assessing fingerprint effectiveness, and we show how it can be used in practice to diagnose the weaknesses in a fingerprint design.

Index Terms— audio fingerprint, copy detection

1. INTRODUCTION

Audio fingerprinting refers to the process of extracting a robust, compact representation of audio which can be used to uniquely identify an audio segment. One major application of this technology is copy detection, where the goal is to detect when a user illegally uploads a television show, movie, or song to a filesharing website. Another application of this technology is music identification, where a user sitting in a restaurant would like to identify a song that is playing in the background. Using a phone application like Shazam [1] or SoundHound [2], the user can record a short audio segment of the song and find out the name of the song and artist.

Many approaches have been proposed for this problem, and here we describe some of the most prominent works. The most well known approach is the Philips fingerprint described by Haitsma and Kalker [3]. The Philips fingerprint considers 33 logarithmically spaced bands below 2kHz and performs 32 comparisons per frame, where each comparison considers whether the energy difference in adjacent frequency bands increases or decreases in two consecutive frames. Ke et al. [4] extends this work by considering the spectrogram as an image and introducing a boosting algorithm to automatically select 32 features (and their corresponding thresholds) from a family of Viola-Jones face detection features [5]. The selected 32 features are computed at each frame and compared to their corresponding thresholds to generate 32 bit fingerprints. Another well known approach is the Shazam fingerprint described by Wang [6]. The Shazam fingerprint identifies the location of spectral peaks in the spectrogram, considers various pairings of spectral peaks, and encodes each peak pair as a single 32 bit fingerprint of the form $(f1, f2, \triangle t)$, where $f1$ and $f2$ represent the frequency of the 2 peaks and $\triangle t$ represents the time difference between the peaks. Another peak-based approach is the waveprint described by Baluja and Covell [7][8]. The waveprint considers sections of the spectrogram as an image, computes the Haar wavelet, identifies the top t wavelet coefficients, and encodes the locations of these maxima with a Min-Hash algorithm.

The above works – and virtually all the works in the audio fingerprinting literature – generally measure the effectiveness of their approach in one of two ways. One way is to consider a fixed searchable database and to measure the recognition rate. This might include the percentage of queries that were identified correctly or indirect measures like the fingerprint bit error rate. The works by Baluja and Covell [7][8], Wang [6], and Haitsma and Kalker [3] use this metric. The second way is to treat the evaluation as a detection problem and measure the tradeoff between false alarms and miss detections (or some derivative thereof). Common metrics include the receiver operation characteristic (ROC) curve, detection error tradeoff (DET) curve, precision-recall curve, normalized cost detection ratio (NDCR), and accuracy at a fixed false alarm rate. Most of the works from the trecvid content based‘ copy detection (CBCD) evaluation [9] fall into this category, since the standardized metric of evaluation is NDCR.

These metrics are very useful in measuring overall system-level performance. However, they are not very useful in measuring fingerprint effectiveness. The systems being measured are usually very complex. The overall system performance reflects many different factors including fingerprint design, the nature of the search mechanism, the amount of redundancy built into the search mechanism, how large the searchable database is, how similar items in the database are to one another, the nature of the noise and distortion, and the configuration of parameter settings. One consequence of this is that it is very difficult to have an intuition about how good or bad a fingerprint design actually is based only on overall system performance.

In this work, we propose a metric that directly measures the effectiveness of a fingerprint design given a set of corresponding clean and noisy audio data. This metric does not require one to implement a search mechanism or construct a database of audio fingerprints, and so it can be computed very efficiently with minimal setup and resources. The metric is simple, easy to compute, and has a clear interpretation from an information theory perspective.

In the next section, we introduce the metric and provide motivation and intuition. To demonstrate the usefulness of this metric, we then apply this metric to the well known Philips fingerprint to diagnose its weaknesses. Having immediate feedback about the effectiveness of a fingerprint design allows a researcher to quickly build intuition and understanding of good design principles with a mini-

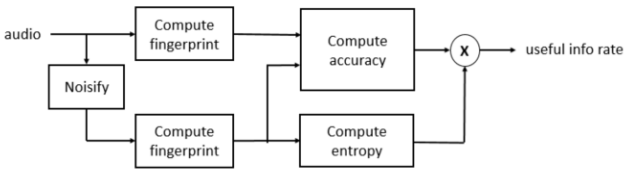

Fig. 1. Block diagram describing computation of useful information rate.

mum of setup and computation.

2. THE METRIC

We propose a metric called useful information rate. It is given by the following simple formula:

$$\text{useful information rate} = \text{entropy} \times \text{accuracy}$$

Here, entropy refers to the number of bits of information that each fingerprint communicates on average. Accuracy refers to the percentage of time that the clean and corresponding noisy audio frames yield the same fingerprint value. To gain some intuition, consider a fingerprint which is all 0's all the time. This fingerprint will have 100% accuracy (since it will always be correct, regardless of how severe the distortion is) but have 0 bits of entropy, resulting in a useful information rate of 0 bits per fingerprint (bits/fp). On the other hand, consider a fingerprint which consists of 32 random, independent bits. In this case, the fingerprint will have an entropy of 32 bits but an accuracy of 2^{-32}, resulting in a useful information rate of approximately 0 bits/fp. This metric has a very clear interpretation from an information theory perspective: each fingerprint communicates a certain amount of useful information (i.e. correct information) on average, and we would like to maximize that information rate.

Figure 1 shows how the useful information rate can be computed given a set of audio data and a distortion model. The distortion model will depend on the application at hand. For example, if the target application is online copy detection, the distortion model might include conversions between different audio formats, varying levels of compression, frame dropping, and equalization. If the target application is music identification on a smart phone, the distortion model might include room reverberation, GSM encoding, and other additive noise sources. Note that we could just as well use real noisy audio data. The only requirement for computing useful information rate is that we be able to map a noisy audio frame to its corresponding clean audio frame. Given the clean and noisy audio, we then extract fingerprints (using the fingerprinting method that we would like to evaluate). To compute accuracy, we compare each clean fingerprint and its corresponding noisy fingerprint and determine what fraction of the time they match. To compute entropy, we use the standard formula

$$\text{entropy} = -\sum_{x_i} p(x_i) \cdot log_2 p(x_i)$$

where $p(x_i)$ represents our estimate of the fingerprint distribution based on histogram counts. The useful information rate is then simply the product of entropy and accuracy.

There are two things to mention about using this metric in practice. First, note that the notion of accuracy can be defined in a flexible manner. For example, some approaches [3][4] build redundancy into the system by searching not just for exact fingerprint matches, but also fingerprints that are, say, within a Hamming distance of 1. In this case, our notion of accuracy becomes "what fraction of clean and noisy fingerprint pairs are within a Hamming distance of 1 of each other?" The definition can be flexible to accommodate the system design, while still retaining a clear interpretation. Second, note that having sufficient statistics to estimate entropy requires exponentially more data as the size of the fingerprint grows. Fortunately, in practice we observe that estimating the entropy with a smaller number of fingerprint bits shows a predictable relationship between the number of bits and the entropy (assuming a consistent method is used to compute additional bits). For example, we observed a very strong linear relationship between the number of bits and entropy in the Philips fingerprint model (where we only use a subset of the bits). So, for a higher number of fingerprint bits, we could simply use a linearly extrapolated estimate of entropy.

Now that we have introduced what the metric is, we now turn our attention to demonstrating how a researcher would *use* this metric to diagnose the deficiencies in a fingerprint design. We will demonstrate this concretely by showing a series of simple experiments that illuminate the inner workings of the Philips fingerprint. There are three main components to its design: the number of bits, the filter design, and the thresholds for each filter. We will investigate these three components in the next 3 sections.

3. FINGERPRINT SIZE

In this section, we investigate how the number of bits affects the effectiveness of the fingerprint. The original Philips fingerprint has 32 bits, where the i^{th} bit represents whether the difference in energy between frequency bands i and $i+1$ increases or decreases between 2 consecutive frames. Here, we investigate how effective the fingerprint is when using only a subset of the bits.

We adopt the general setup shown in figure 1, using audio data from the trecvid CBCD task. For the sake of simplicity and clarity, we will consider a simple distortion model which adds white gaussian noise at various signal-to-noise ratios (SNRs). This setup will allow us to characterize the performance across a range of conditions rather than just getting a single performance metric. To emphasize how lightweight the setup can be and yet still be useful, we will compute the useful information rate on 10,000 randomly sampled frames (and their surrounding context), rather than on the full 400 hour trecvid data set. This corresponds to only a few minutes of audio.

Figure 2 shows the useful information rate of the Philips fingerprint with l bits, for l ranging from 1 to 32. We see that using more bits would be only marginally helpful in the high SNR regime. At 30dB SNR, the useful information rate has mostly leveled off by the time l reaches 32. For the medium SNR range (10-20dB), we see that the useful information rate peaks between 8 and 15 bits. We can use significantly fewer bits and get both better performance and less computation. For the low SNR range ($<$10dB), we see that using 32 bits is a very poor choice of l.

Note that the results in figure 2 are the result of two conflated factors: the number of bits and the amount of spectrum that is considered. Even so, our analysis indicates that the Philips fingerprint probably has way too many bits. For most applications, its performance would improve if fewer comparison were performed. In the literature, there is often an unspoken assumption that squeezing a fingerprint into a single 32 bit representation is good because it means the fingerprint designer was economical with memory. The reality, however, is that using 32 bits is probably way too high for a

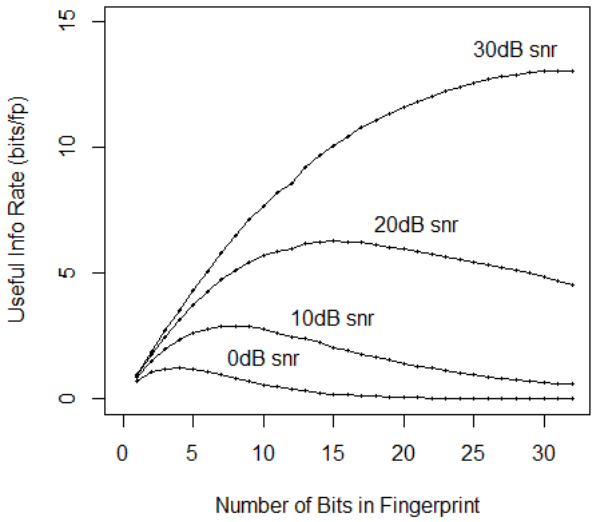

Fig. 2. Relationship between fingerprint size and useful information rate.

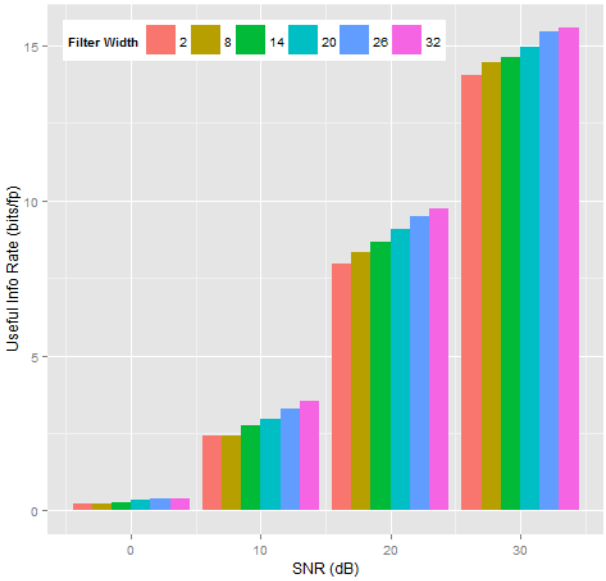

Fig. 3. Effect of filter width on useful information rate.

threshold-based approach like the Philips fingerprint. Note that a set of 32 perfectly uncorrelated fingerprint bits with impressive accuracies of .97 would only have a fingerprint accuracy of approximately .38.

4. FILTER SELECTION

In this section, we investigate the question of filter design. The original Philips fingerprint uses a 2×2 checkerboard filter centered at different frequency bands. But one might wonder if there is a better selection of filters.

Consider one of the 2×2 checkerboard filters. The fingerprint bit corresponding to this filter is determined by adding and subtracting the appropriate values and then comparing the result to the threshold value of 0. By its symmetrical construction and the law of large numbers, we know that the feature (before being thresholded) will have a bell-shaped distribution centered at 0. If a particular feature falls close to 0, a small amount of perturbation from noise may cause it to fall on the wrong side of the threshold, which would result in an incorrect bit. We can minimize this occurrence by increasing the spread of the feature distribution. This suggests that we can improve fingerprint effectiveness by selecting filters that maximize the variance of the feature distribution.

We can very quickly test this hypothesis by evaluating several fingerprint designs on our small sample of 10,000 frames (along with their context frames). Since "pixels" in the spectrogram that are close together in time or frequency will tend to be highly correlated, we want to avoid taking the difference of adjacent elements. So, instead of the 2×2 checkerboard filters in the original Philips model, we consider a set of $2 \times m$ filters in which the leftmost $\frac{m}{2}$ columns are added and the rightmost $\frac{m}{2}$ columns are subtracted. As before, the i^{th} filter will be centered at frequency band i. As m increases, we expect the variance of the feature distribution to increase, and thus for the fingerprint to be more robust. Is this what we observe?

Figure 3 shows the result of these experiments. Each group of bars shows the useful information rate of the fingerprint model with width m at a given SNR. Indeed, we observe that as m increases, the fingerprint model improves slowly but steadily. This validates the idea that maximizing the variance of the features will yield the most robust fingerprint. In a sister submission [10] we take this idea to its logical conclusion and formulate the fingerprint design as an optimization problem which maximizes variance. Ironically, this experiment also indicates that the 2×2 filters in the Philips fingerprint is one of the *worst* possible selections of filters, since it only adds and subtracts immediately adjacent pixels!

The useful information rate metric matches our understanding of good fingerprint design. But does it also match system-level metrics of performance?

Figure 4 shows the system-level accuracy for the same experiment. Here, we created a database of fingerprints for 977 files totaling approximately 40 hours of audio data. We generated 500 noisy queries by randomly selecting 10 second segments and adding noise at a desired SNR. To compute a score for each item in the database, we used the search method described by Wang [6]. The accuracy in figure 4 refers to the percentage of queries that were identified correctly (i.e. the true match had the highest score). This is one of the common metrics used to measure system-level performance, as described in section I.

There are two things to notice when we compare figures 3 and 4. First, we notice that useful information rate and overall system-level accuracy correlate directly when assessing the *relative* performance of various fingerprint designs. Within each grouping of bars (in both figures), the performance improves as the filter width increases. In this way, we can compare fingerprint designs more efficiently using useful information rate rather than system-level performance. Note that figure 4 required more than 1000 times more data and significantly more setup and implementation than figure 3. Second, we point out that these two metrics do *not* correlate directly in terms of *absolute* performance. For example, notice that a fingerprint filter width of 32 at 10dB SNR has higher system accuracy than a finger-

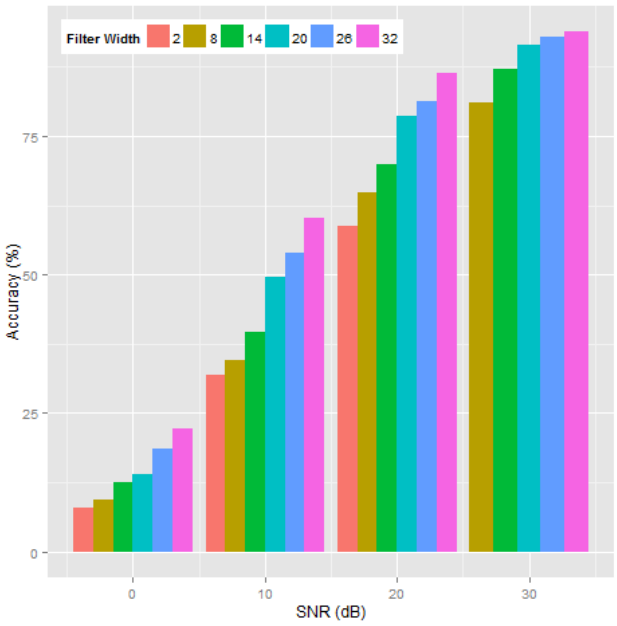

Fig. 4. Effect of filter width on system-level accuracy.

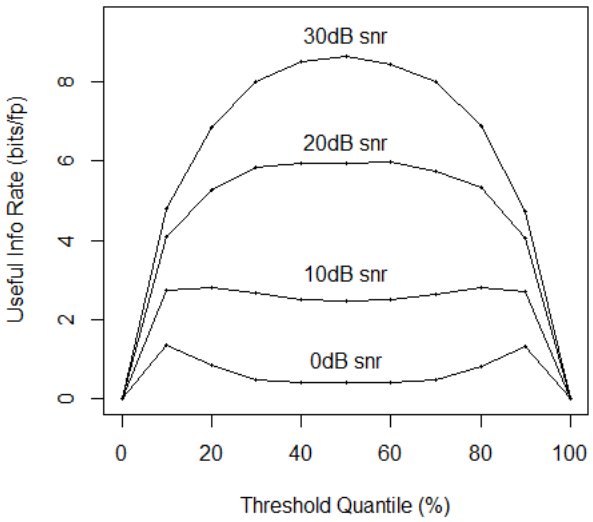

Fig. 5. Effect of threshold selection on fingerprint effectiveness.

print filter width of 2 at 20dB SNR. In contrast, a fingerprint filter width of 32 at 10dB SNR has a much *lower* useful information rate than a fingerprint filter width of 2 at 20dB SNR. Again, we point out that the absolute performance numbers depend upon many factors besides fingerprint design, such as the database size and the amount of redundancy built into the search mechanism.

Figures 3 and 4 demonstrate that useful information rate and system-level accuracy correlate directly in evaluating fingerprint designs. The useful information rate metric is preferable because it requires much less data and almost no setup to compute.

5. THRESHOLD SELECTION

We have examined the number of bits and how to compute each bit. We now consider the last remaining piece of the design puzzle: the thresholds. In the original Philips fingerprint, the threshold is set to 0 (i.e. the result of adding and subtracting pixels is compared to 0 in order to determine the bit value). Intuitively, it seems that we should set the threshold to the median in order to maximize entropy.

Again, we can quickly test this hypothesis by evaluating variants of the original Philips fingerprint design on our small sample set. These variants all use the same original set of 2×2 checkerboard filters, but are thresholded at different quantiles of the feature distribution (where, for example, the 50% quantile corresponds to the median).

Figure 5 shows the results of this experiment. Each curve shows the useful information rate of the various designs at a given SNR. We see that at high and medium SNRs, the median does indeed maximize the useful information rate, as our intuition suggests. However, we notice that at low SNRs, the useful information rate with a median threshold actually becomes the *worst* of all the other designs.

Why would you ever set your threshold to something other than the median? These experiments suggest that there are situations when it is better to set your threshold differently. When the noise is really bad, there is no point in trying to maximize the entropy anymore, since the accuracy is too low. In these cases, it is better to sacrifice entropy in order to boost accuracy, and this can be achieved by moving the threshold towards one of the tails of the distribution. The feature will fall on one side of the threshold most of the time (resulting in low entropy), but when it does occasionally fall on the other side of the threshold, it is more likely to suggest that the underlying (clean) feature value actually does fall in the tail of the distribution. So, our experiments do not change our design decision of setting the threshold to the median, but it does provide more insight into the factors at hand.

6. CONCLUSION

We have proposed a metric called useful information rate that directly measures the effectiveness of an audio fingerprint design on a set of clean and noisy audio. We show that with minimal setup, effort, and computation, the metric can be used to quickly assess the fingerprint's performance. Useful information rate has a very clear and satisfying interpretation from an information theory perspective, and we demonstrate that it correlates directly with system-level performance in assessing fingerprint effectiveness. Providing this immediate feedback allows a researcher to iterate quickly and build greater understanding and intuition of good fingerprint design.

7. REFERENCES

[1] "Shazam," Available at `www.shazam.com`, Oct 2014.

[2] "SoundHound: The most immersive music search, discovery and play experience on mobile," Available at `www.soundhound.com`, Oct 2014.

[3] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2002, pp. 107–115.

[4] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, vol. 1, pp. 597–604.

[5] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[6] A. Wang, "An industrial strength audio search algorithm," in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2003, pp. 7–13.

[7] S. Baluja and M. Covell, "Audio fingerprinting: Combining computer vision data stream processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2007, vol. 2, pp. 213–216.

[8] S. Baluja and M. Covell, "Waveprint: Efficient wavelet-based audio fingerprinting," *Pattern Recognition*, vol. 41, no. 11, pp. 3467–3480, 2008.

[9] P. Over, G. Awad, J. Fiscus, B. Antonishek, M. Michel, A. Smeaton, W. Kraaij, G. Quénot, et al., "TRECVID 2011-an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *TRECVID 2011-TREC Video Retrieval Evaluation Online*, 2011.

[10] T. Tsai, E. Chu, and G. Friedland, "Learning audio fingerprints via projected eigenvectors," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, In submission.