# Analysis of Minimum Distances in High-Dimensional Musical Spaces

Michael Casey, *Member, IEEE*, Christophe Rhodes, and Malcolm Slaney, *Senior Member, IEEE*

***Abstract*—We propose an automatic method for measuring content-based music similarity, enhancing the current generation of music search engines and recommender systems. Many previous approaches to track similarity require brute-force, pair-wise processing between all audio features in a database and therefore are not practical for large collections. However, in an Internet-connected world, where users have access to millions of musical tracks, efficiency is crucial. Our approach uses features extracted from unlabeled audio data and near-neigbor retrieval using a distance threshold, determined by analysis, to solve a range of retrieval tasks. The tasks require temporal features—analogous to the technique of shingling used for text retrieval. To measure similarity, we count pairs of audio shingles, between a query and target track, that are below a distance threshold. The distribution of between-shingle distances is different for each database; therefore, we present an analysis of the distribution of minimum distances between shingles and a method for estimating a distance threshold for optimal retrieval performance. The method is compatible with locality-sensitive hashing (LSH)—allowing implementation with retrieval times several orders of magnitude faster than those using exhaustive distance computations. We evaluate the performance of our proposed method on three contrasting music similarity tasks: retrieval of mis-attributed recordings (fingerprint), retrieval of the same work performed by different artists (cover songs), and retrieval of edited and sampled versions of a query track by remix artists (remixes). Our method achieves near-perfect performance in the first two tasks and 75% precision at 70% recall in the third task. Each task was performed on a test database comprising 4.5 million audio shingles.**

***Index Terms*—audio shingles, distance distributions, locality-sensitive hashing, matched-filter distance, music similarity.**

## I. Introduction

We ARE interested in efficient algorithms for finding similar musical pieces based on their acoustic signal. Systems for song retrieval are most clearly characterized by the specificity of the query. To illustrate, let us consider four types of audio queries; the first three types have the common property that they match sequences of audio features; therefore, they use temporal features, while the fourth and least specific can be answered using statistical averages.

1) Fingerprinting: Audio fingerprints are the basis of the most specific type of query. The fingerprint query aims to uniquely identify a recording in the presence of a distorting communications channel. For example, a song might be corrupted by loud background noise and very lossy compression. Fingerprinting is valuable to consumers as it allows delivery of more information about a song they are hearing.
2) Remixes: A remix is a version of a song that contains audio content from the original song, but it is remixed to give a different feel, often but not always by the same artist. A remix query returns songs which contain a (possibly distorted but closely matching) fragment of the query song.
3) Cover songs: In contrast, cover-song retrieval seeks to find all versions of the same song title performed by different artists. In this case, we do not expect any of the query's audio to be contained within the retrieved songs, i.e., it is the overall *musical* content that is in some sense similar.
4) Genre: The least specific type of audio matching disregards temporal information altogether and attempts to match global spectral properties of songs, usually using temporally insensitive statistical models.

In fingerprinting, we typically look for exact matches, but the second and third types of queries are approximate matches, and thus we need to find near neighbors in some feature space.

Whether a song is relevant to a given query type and query track can be determined by counting the number or proportion of near-neighbors, short sequences of feature vectors from the query track, which are from relevant songs. This paper addresses the problem of finding near neighbors for acoustic signals in a high-dimensional space, as well as the issue of determining a relevant threshold for such neighbor computation.

Potential applications include near-duplicate elimination for improving the ranks of relevant search results, sampling source identification, high-level music structure extraction (identifying repeats), remixed-song retrieval, cover-version retrieval, and linking of recommendation data between closely related songs. We wish to describe an implementation of such applications in today's million-song music databases.

### A. Approach

*1) Audio Shingles:* Our approach is motivated by the specificity of our queries and by the scale of today's music search problems. Specific queries require sequences of features for similarity evaluation, rather than either single feature or bag-of-features models. We organize a sequence of feature

Manuscript received December 17, 2006; revised March 18, 2008. This work was supported by the Engineering and Physical Sciences Research Council under Grants EPSRC GR/S84750/01 and EPSRC EP/E02274X/1. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vesa Valimaki.

M. Casey is with the Department of Computing, Goldsmiths College, University of London, London SE14 6NW, U.K., and also with the Department of Music, Dartmouth College, Hanover, NH 03755 USA (e-mail: mcasey@dartmouth.edu).

C. Rhodes is with the Department of Computing, Goldsmiths College, University of London, London SE14 6NW, U.K. (e-mail: c.rhodes@gold.ac.uk).

M. Slaney is with Yahoo! Research, Inc., Santa Clara, CA 95054 USA (e-mail: malcolm@ieee.org).

Digital Object Identifier 10.1109/TASL.2008.925883

vectors $f_i$ with dimension $m$, into a shingle of dimension $M = l \times m$

$$a_i = [f_i, f_{i+1}, \dots, f_{i+l-1}]. \tag{1}$$

Shingles are taken exhaustively from each track by advancing the shingle window by one feature vector.

*2) Track Model:* Our model of between-track similarity counts matching shingles. Thus, pairs of tracks with more matching shingles are more similar. To determine when two shingles are a match, we use a distance threshold: $x_{\text{thresh}}$. We are only interested in those pairs of shingles with distance falling below this threshold, see Section III-C. Thus, we express the track similarity problem as a radius-bounded, near-neighbor search which, as discussed below, allows efficient implementations.

*3) Efficiency:* The shingling technique that we employ for temporal specificity in our queries multiplies the dimensionality of features by a shingling factor l, see (1). For low-dimensional data, there exist many space-partitioning methods that solve approximate near-neighbor search with sublinear time complexity. Locality-sensitive hashing (LSH) algorithms [1], on the other hand, are based on random projections of high-dimensional features to low dimensions and solve radius-bounded approximate near-neighbor problems with sublinear time complexity. Our method is directly compatible with such algorithms and therefore can be efficiently implemented using LSH.

*4) Null Hypothesis:* We determine the distance threshold from unlabeled audio data; that is, we sample the between-shingle distances to estimate a background distance distribution. This forms the null hypothesis: that the distance between two shingles is drawn from the background distance distribution. When the probability of the distance between shingles is in the tail of the distribution, tending toward zero distance, we reject the null hypothesis and conclude that the distance between the pair of shingles is due to another distribution: that of matching shingles. This requires a robust method to estimate the distance value, and therefore a radius where we reject the null hypothesis. This is achieved by analysis. First, we derive a model of the between-shingle background distribution and from this we derive a model of the distribution of minimum values of the null-hypothesis distribution. The limiting tail of the value distribution contains the shingles that match, see Fig. 1.

*5) Specificity:* The specificity of shingle matching for music querying depends on the features and the distribution of between-shingle distances in the database. For example, a database that consists of many versions of the same recording, with different distortions, will have a narrower background distance distribution that is weighted closer to zero than a database consisting entirely of different tracks by different artists in different musical styles. Those pairs of shingles whose distance do not fit the background will differ in the two databases because the probability distributions suggest a different distance for rejecting the null hypothesis. By selecting different parts of a database to compute a background distribution, or by changing the feature from which shingles are drawn, we solve different queries by sampling to form a null hypothesis and rejecting this hypothesis for matching pairs.

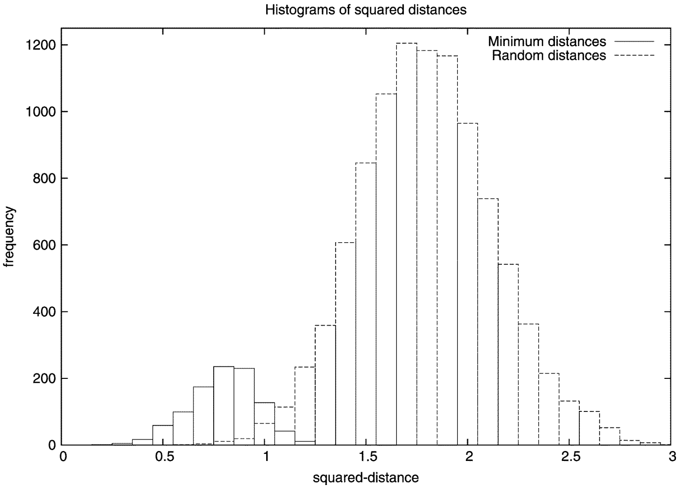

Fig. 1. Squared-distance histograms of two distributions from a database of 1.4 million shingles. The left bump is the minimum between 1000 randomly selected query shingles and this database. There are few matches. The right bump is a small sampling (1/98 000 000) of the full histogram of all distances and illustrates the difference in location and dispersion of the two distributions. The task-dependent similar shingles are in the left tail, toward zero, of a very large distribution.

*6) Evaluation:* We evaluate our proposed method on three contrasting music similarity tasks: retrieval of misattributed recordings (fingerprint), retrieval of the same work performed by different artists (cover song), and retrieval of edited and sampled versions of a query track by remix artists (remixes). The three tasks form a continuum of high- to mid-specificity music queries, thus demonstrating that our method solves a range of music-similarity tasks.

*7) Contributions:* The contributions in this paper are: 1) between-track similarity expressed as radius-bounded, near-neighbor retrieval; 2) a new algorithm for similar-track retrieval that admits efficient implementation using LSH; 3) analysis of between-track distance distributions and minimum-value distance distributions; and 4) experimental evidence that the track-similarity problem, as stated, is well-posed.

*8) Structure:* The structure of the paper is as follows. We review different approaches to audio similarity in Section II; we develop a model for retrieving tracks by audio shingles in Section III-B; we then give details of related-track recognition algorithms in Section III-C and give an analysis of the algorithms in Section IV; finally, we present the results of various related-track retrieval experiments on 4.5-million feature vector data sets in Section V, and we conclude with the implications of our results for efficient audio matching applications in Section VI.

## II. Related Work

There is a wide spectrum of previous work covering a range of music-similarity tasks: from very specific identification or fingerprinting work [17], [29] to genre recognition [28], [24]. Our work falls in the middle: we aim to find songs that are similar, but not exactly like another song. Our tasks need both musically relevant features—we do not expect the very specific features used in fingerprinting to work—and a robust matching criterion, because we expect that remixes will have segments rearranged and new material inserted.

All fingerprinting and musical-similarity systems combine feature analysis and a matching stage. There is always a tradeoff between the complexity of the feature and the amount of effort needed to discriminate good from bad matches in the song database. Roughly speaking, each system transforms the audio into a musically relevant feature vector, then looks for that vector in the music database. Both of these steps are hard: the wrong feature magnifies meaningless differences between the query and the database entries, and brute-force, linear-search techniques are no longer viable due to the size of today's music libraries. For context, we will talk about each aspect in turn, as they are used in other music-similarity systems.

### A. Fingerprinting

Querying for similar songs is different from the classic work that has been done on audio fingerprinting [9], [19], [22], and [29]. These fingerprint systems typically assume that some portion of the audio is an exact match such that conventional computer hashes can reduce the search space. We do not expect to see exact matches in remixes.

Our work also differs from traditional fingerprinting or matching work because we choose a different division of labor between the feature and the matching stages. Many systems use relatively short features, perhaps 20 ms long, and then use a more complicated matching scheme to find a sequence of matches. Our approach, on the other hand, uses a long acoustic-feature vector (3 s) and a single, conceptually simple, lookup algorithm to find the near neighbors. We implement the near-neighbor lookup using a randomized algorithm, LSH, which we believe is of increasing importance as our databases grow in size. The long feature vector and the relatively simple lookup algorithm makes analysis of our performance feasible.

### B. Features for Similarity

Features used in music-similarity tasks range from simple to specific. At the simplest level, features provide a relatively straightforward representation of the musical signal. Examples of such systems include Kurth [23] and our own work on the cover song and remix tasks [10], [11]. Both these systems use a musical representation based on 12 equal-temperament chroma in an octave. With the exception of smoothing or downsampling of the signal, the resulting feature vector contains the essential aspects of the musical signal. One could reconstruct a musically expressive audio signal from these features alone.

The same is not true of the classic fingerprint systems. In fingerprinting, users want to find the name of a recording, given a sample of the audio. The key to make fingerprinting work is the use of a robust feature of the signal that is not harmed by difficult communications channels (e.g., a noisy bar or a cell phone). Typical features include the peaks of a spectral–temporal representation [29], the relative energy levels in adjacent spectral bands [18], and the spectral–temporal wavelets with maximum energy [2]. These aspects of the signal are not (often) harmed by noisy environments and are good for finding exact matches.

More recent fingerprinting or identification systems have learned their feature representation. Burges [8] used an optimization procedure to design acoustic features that are robust to nine low-level distortions: compression, nonlinear amplitude/bass distortion, various notch filters or frequency boosts, pitch distortion, and companding. They learn projection vectors by training the system with noisy and clean versions of the same signal, and from all these vectors they estimate suitable overall covariance and correlation matrices. Similarly, other researchers have learned features from a timbral representation known as Mel-frequency cepstral coefficients (MFCCs) [30] or by optimizing their features based on spectrograms as part of an overall identification system [3], [30]. The adaptive nature of these features makes it hard to analyze their performance.

### C. Querying for Similarity

Given a feature set, there are many ways to find the database entries that are closest to the query. The brute-force solution is easy to describe: look at each database entry, compute the pairwise distance, and then return the pair with the minimum distance. In practice, query approaches differ based on whether the algorithm finds exact matches, or whether they allow approximations, and whether they are concerned with large databases and need efficient solutions.

Brute-force techniques, searching every database entry, are possible for small databases. Mueller's audio matching system [23] is successful with this approach, as well as Burges' [8]. However, this will not work in practice for databases with millions of songs containing billions of audio snippets.

Conventional hashing techniques work well if one is looking for an exact match. A conventional hash converts a set of data, whether a string or a numerical vector, into an index into a table. If the desired entry is present in the table, it is returned; otherwise, a miss is recorded. This converts an $O(N)$ problem, where $N$ is the size of the database, into an $O(1)$ problem in time, but $O(N)$ in memory. This type of exact hash is the basis of the classic fingerprinting work [18], [29]. Ke [21] extends their low-dimensional queries to include nearby variations based on small, deterministic changes to the query vector.

When using larger databases, and for systems that are interested in similarity as opposed to identification, a more sophisticated approach is necessary. LSH was originally designed to find nearby web pages [7], but has since been extended to music [31], image retrieval, and other tasks [15]. At its core, LSH is a randomized algorithm that finds near neighbors in high-dimensional spaces [26]. In practice, it is an implementation technique—it approximates the best answer in sublinear time. It is a practical solution to the problems described in this paper, and others [2]. Baluja [3] describes an improvement of LSH that replaces the random projections in LSH with "projections" that are learned from loose labels on the data.

Finally, a new approach by Weinstein [30] builds a finite-state automaton to recognize a large collection of music. They train up to 1024 musical "phones" by clustering, and then recognize sequences of these phones using the automata. A collection of 15 000 songs, after combining related nodes, leads to a finite-state automata with 60 million edges.

Our work in this paper uses two simple features (to be defined in Section III-A) and our matching threshold is motivated by LSH's query threshold. The simple features and the brute-force query allow us to perform the statistical analysis we describe in Section IV. Employing this analysis, we chose a theoretical optimum radius for a given false alarm rate in retrieving similar audio sequences. This paper focuses on estimating and testing the optimum radius in three different tasks.

### D. Query Optimization

Finally, we are not the first to note that nearest neighbors is a difficult problem in high dimensions. Beyer shows that the problem becomes unstable as the dimensionality tends toward infinity because the nearest neighbor and the farthest neighbor are so close in distance [5]. In this paper, we show the problem is not so bleak with realistic data and finite dimensions. Beyer's work led to important work characterizing various indexing schemes that might or might not be useful in high-dimensional spaces [25]. Shaft notes the sensitivity of LSH to the search radius, a problem we mitigate with the statistical analysis in this paper because we derive statistical properties of our high-dimensional musical spaces and use them to derive the best search radius.

Query optimization is an important part of modern database systems—a database system wants to estimate how much data different parts of a complicated query are going to return so they can properly order the calculations for most efficient processing. Thus, depending on the computational complexity, one needs to decide whether to first search for nearest neighbors or for some other aspect of the query. The cost of searches and the expected number of results for queries in multidimensional spaces have been addressed, both for low- and medium-dimensional spaces [27] and high-dimensional spaces [6]. We continue this line of work by deriving an estimate of the true dimensionality of the data, and derive the analytical expressions for the probability distributions of nearest neighbors.

## III. Track Model

In this section, we develop our track model starting from audio features, then describe how we perform audio shingling and, finally, discuss the distance measure and similarity model for two tracks. The motivation for the methods described in this section was presented in Section I-A.

### A. Audio Features

We look at two different features in this work. We chose: log-frequency cepstral coefficients (LFCCs) for the high-specificity task (fingerprint retrieval); and equal-temperament pitch-class profiles (PCPs) for cover-song retrieval and remix retrieval, the mid-specificity tasks.

All features are extracted from uncompressed audio sourced from the CHARM Mazurkas collection [13], and the Yahoo! Music database. The audio sources are in 44.1 kHz uncompressed, stereo format, which we convert to mono by extracting the left channel for the fingerprint task and by mixing the channels for the other tasks. We window the audio using a 8192-point Hamming window with 4410-sample (100 ms) hop and compute a 16384-point discrete Fourier transform (DFT) using the fast Fourier transform (FFT). We choose the FFT window size so that two samples are available from the DFT for the lowest pitch class surrounding $C0 = 65.4$ Hz; i.e., DFT samples at 64.59 Hz and 67.29 Hz. The band edges are chosen at the mid-point (quartertone) between chromatic pitch classes with the low-edge set to 63.5 Hz and high-edge of 7246.3 Hz, a quartertone above A7.

We assign the 8193 DFT magnitudes to pitch-class bands and share samples near the band edges proportionally between the bands [4], and we ignore the remaining DFT coefficients. We then fold the pitch-class assignments into a single octave by summing over all octaves for each pitch class, and take the logs of sums yielding the 12-dimensional PCP vector every 100 ms.

The LFCC features used the same logarithmic bands as the PCP but instead of folding down into one octave we take the log of the values and transform the result to cepstral coefficients using a discrete cosine transform (DCT) yielding 20 coefficients per 100 ms. Note this feature is a slightly modified form of the widely used MFCC feature, the difference being in the concentration of low-frequency components. We designed this feature so that we can reuse the FFT coefficients between the PCP and LFCC calculations.

### B. Audio Shingles

By analogy to the work on finding duplicate web pages [7], audio shingles represent short (several seconds) segments of audio. We concatenate 30 feature vectors at a time into a single high-dimensional vector representing a sequence of audio features. Informed by previous studies [11], [23], we use a window of 3 s with a hop of 0.1 s yielding 10 Hz $\times$ 20 dimensions $\times 3$ s $=$ 600 dimensions for LFCC and 10 Hz $\times$ 12 dimensions $\times 3$ s $=$ 360 dimensions for PCP.

We did not want silence or noise to be included in the matches between songs, since these two processes are generic to all songs and might corrupt the recognition of similar content. We removed silence by thresholding audio segments by one quarter of the geometric mean of the shingle power in each song.

The remaining shingles are normalized to unit norm so that differences in power between pairs of shingles are not included in the distance metric. This provides invariance to differences in levels between different recordings of the same work, for example, or different treatments of the same track, as in the fingerprint task.

### C. Track Similarity

Let $\{Q\}$ be the set of $M$-dimensional audio shingles from a query sound and $\{A^{(n)}\}$ be the shingles for track $n$ (an entire song) drawn from the database with $N$ tracks. Let $q_i \in \{Q\}$ and $a_j^{(n)} \in \{A^{(n)}\}$ be shingles drawn from tracks $\{Q\}$ and $\{A^{(n)}\}$, respectively. We define the similarity between tracks using the Euclidean distance implemented by a dot product between the two unit-normed shingle sequences. This operation is equivalent to a multidimensional matched filter, which we see by expanding the quadratic formula for distance and noting that the two vectors are unit norm

$$x = d^2\left(q_i, a_j^{(n)}\right) = 2 - 2\sum_{m=1}^{M} q_{im} a_{jm}^{(n)}. \tag{2}$$

Then, we measure the similarity between the query track and a database track $n$ by counting the number of query shingles $q_i \in \{Q\}$ that are within the distance threshold $x_{\mathrm{thresh}}$ of the track's shingles $a_j^{(n)} \in \{A^{(n)}\}$. We do this by first forming a binary indicator function $I_i(q_i, A^{(n)}) \in \{0, 1\}$

$$I_i\left(q_i, A^{(n)}\right) = \begin{cases} 1, & \text{if } \exists j \text{s.t. } d^2\left(q_i, a_j^{(n)}\right) \le x_{\text{thresh}} \\ & \text{for } i \in \{1 \cdots |Q|\} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $|Q|$ denotes the cardinality of the set $Q$. Finally, we define the similarity count for the track pair $(Q, A^{(n)})$ to be

$$C\left(Q, A^{(k)}\right) = \sum_i I_i\left(q_i, A^{(n)}\right). \tag{4}$$

The indicator function of (3) tells us whether any parts of the two tracks are similar, so that only one relevant shingle in each track is counted per query shingle. The maximum count, therefore, is $|Q|$ and the minimum is 0. Those tracks from the database with the highest count, with respect to the query track, are the most similar. The value $x_{\text{thresh}}$ is critical. Values of $x_{\text{thresh}}$ that are too small result in retrieval of too few shingles and the false negative rate for retrieving near shingle pairs is high. In the extreme $x_{\text{thresh}} \approx 0$, no shingles are returned for all tracks. For values of $x_{\text{thresh}}$ that are too large, the false positive rate will be high, admitting pairs of shingles as being relevant where they are not. In the extreme for $x_{\text{thresh}} \approx 4$, all tracks in the database return a count equal to the number of shingles in the query track.

For application to automatic retrieval systems, we propose a method to estimate the optimal value of $x_{\text{thresh}}$ given a sample of the data set, so that the proportion of near shingles is higher for relevant tracks than nonrelevant. In Section IV, we present an analysis of the statistics of distance distributions to derive a set of equations to estimate $x_{\text{thresh}}$ given a sample of nonrelevant shingle distances. These methods of computing inter-track similarity have the following desirable properties.

1) We have an analytic method to derive a decision boundary for detecting relevant and nonrelevant shingle pairs for a retrieval task. The method uses a sample of nonrelevant pairs of shingles as training data.
2) We have a measure of the resemblance between two tracks that is robust to structural changes such as repeats, insertions, and deletions. This is necessary for cover song and remix retrieval, where versions of a work exhibit global structural differences but preserve the local sequence information of the work.
3) There exists an efficient means to solve the above distance and track ordering expressions. In naive implementations, we evaluate the distances between the query shingles and all database entries, and the resulting time complexity is $O(Nd)$ with respect to the total number of shingles $N$ in the database and shingle dimensionality $d$. LSH retrieves only those database shingles that are within $x_{\text{thresh}}$ of the query shingles, thus we can evaluate (3) and (4) with sublinear time complexity.

## IV. Modeling Distances Between Shingles

The distance between two shingles is the Euclidean distance (in this $M$-dimensional space) between the two vectors. These vectors are long, perhaps hundreds of dimensions. It is hard to find near neighbors in this kind of very high-dimensional space without resorting to exhaustive distance computation.

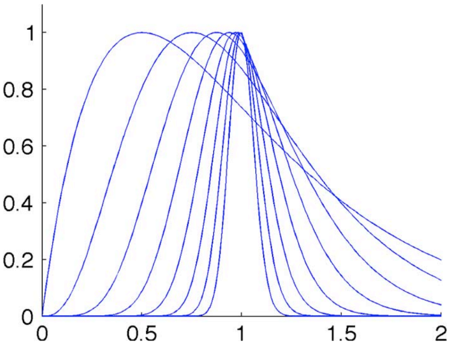

Fig. 2. Histogram of normalized distances for points placed uniformly at random in 4 (left-most curve), 8, 16, 32, 64, 128, 256, and 512 (inner curve) dimensional spaces. As the dimensionality of the space increases, the distribution gets narrower, indicating that most points are equidistant.

Most importantly, the curse of dimensionality [20] means that as the dimensionality grows, most data points are nearly equidistant. This is demonstrated in Fig. 2, which shows a probability distribution function (pdf) of the distance between points placed at random in a series of high-dimensional spaces. As the dimensionality of the feature space grows, the distribution of distances becomes more sharply peaked. This suggests that finding near neighbors in a high-dimensional space might be ill posed [20]—all points are nearly the same distance from each other. We are only interested in the left-hand tail of this distribution, the pairs of points with the smallest distances between them, but still we need to be careful to not look at too many points.

We wish to know if the shingle distances, from similar tracks, are distributed differently from the background of nonsimilar tracks, so that there are enough shingles in the left tail of the distance distribution, so the near-neighbor question is well posed. This analysis is also important because it will help us determine the optimum radius when using LSH to search for similar shingles. In this section, we derive analytic approximations for five quantities related to our problem.

IV-A the pdf of uncorrelated shingle distances.

IV-B an estimate of distribution parameters based on sampling distances to other shingles.

IV-C the pdf of the nearest neighbor in a chi-squared distribution.

IV-D the pdf of the nearest neighbors for audio shingles.

IV-F a decision rule for determining related shingles.

The pdf of uncorrelated shingle distances follows a chi-squared distribution (see Section IV-A). We derive an approximation of this distribution for shingles that are close to each other. We then derive a maximum-likelihood estimate of the distribution parameters, given a set of sample data. Furthermore, we can derive an estimate of the pdf for the distance to the single shingle that is closest to the query. This is what we need to derive a threshold, which is a decision rule, that allows us to decide whether a song's shingles are correlated or uncorrelated to a given a query sample.

### A. PDF of Shingle Distances

Our technique to find related songs is based on short snippets of sound we call shingles. For this analysis, we start with the assumption that each element of the song's feature vector is from a Gaussian distribution. To calculate the distance between two shingles, we subtract two Gaussian variables, square the result, and sum over all dimensions. The magnitude of the difference between two independent, identically distributed (IID) Gaussian random vectors is also Gaussian and the probability density function (pdf) is given by the chi-squared $\chi^2$ distribution.

The shape of this distribution depends on the number of independent dimensions in a shingle. The pdf of distances will be narrow if each frame of data used to create the shingle is independent of the other (temporal) frames. The pdf of distances will be wide if some of the frames used to create the shingle are duplicates of others. This affects our calculation of the tails of the distribution, where the nearest neighbors are found, and motivates the discussion below about the degrees of freedom.

Consider two different extreme cases. First, assume the elements of the shingles are drawn from completely independent Gaussian processes with an overall population mean $\mu$ and variance $\sigma^2$. Then each of the $M$ elements in each shingle is independent of all the others in that shingle, and of all the elements in all other shingles (assuming the shingles are taken with no overlap). By definition, the squared distance between the shingles are distributed as $\sigma^2\chi^2_M$, where $\chi^2_M$ is a random variable consisting of the sum of squares of $M$ Gaussian random variables with zero mean and unit variance. At the other extreme, consider a process where all elements of the shingle are a single identical random variable. Then all of the $M$ elements of each shingle are the same, and the squared distance are distributed as $M\sigma^2\chi^2_1$: one single squared quantity, multiplied by $M$. Note that both of these distributions have the same mean, but the $\chi^2_M$ distribution is much narrower.

In our case, we form the $M$-element feature vector from $M/K$ concatenated frames of $K = 12$ dimensional PCP or $K = 20$ LFCC features. We expect the distribution of the squared distances to lie somewhere between the two extreme distributions above. Not only might some of the frames be duplicated because the acoustic signal might change slowly, there are likely to be less than K degrees of freedom in the PCP of one frame.[1] Successive frames are highly correlated, and so will add few extra degrees of freedom—certainly not as many as K per frame. Denoting the effective number of degrees of freedom as $d$, we hypothesize the squared distances between shingle vectors *drawn from unrelated songs* are a random variable of the form $\sigma^2(M/d)\chi^2_d$.

The pdf for a $\chi^2$ distribution with $d$ independent degrees of freedom is

$$f_\chi(x;d) = \frac{1}{(2^{d/2})\mathbf{\Gamma}(d/2)} x^{(d/2)-1} e^{-x/2}. \tag{5}$$

Therefore, the pdf of the *squared* distance $x$ between two $M$-dimensional shingles, each with $d$ effective degrees of freedom is

$$f_s = \frac{d}{M\sigma^2} f_\chi\left(\frac{d}{M\sigma^2}x; d\right) \tag{6}$$

$$= \left(\frac{d}{2M\sigma^2}\right)^{\frac{d}{2}} \frac{1}{\mathbf{\Gamma}\left(\frac{d}{2}\right)} x^{\frac{d-2}{2}} e^{-\frac{d}{2M\sigma^2}x} \tag{7}$$

where $\sigma^2$ is the variance of the element-by-element distances and $0 < x < \infty$. Here, $d$ is a parameter expressing the internal structure of that space—how correlated different components of the individual feature vectors are with each other, and with the components of other feature vectors.

Near $x = 0$, that is, near the minimum possible distance between two feature vectors, this distribution is approximately a power law. By expanding the exponential in (7) as a power series in $x$ and taking the limit, keeping only the lowest-order term in $x$, we approximate the distance pdf with

$$f_s \approx \left(\frac{d}{2M\sigma^2}\right)^{\frac{d}{2}} \frac{1}{\mathbf{\Gamma}\left(\frac{d}{2}\right)} x^{\frac{d-2}{2}}. \tag{8}$$

The cumulative distribution function (cdf) $F(x)$ is the integral of (8) with respect to $x$, which near $x = 0$ is

$$F(x \text{ for } \|x\| \ll 1) \approx \frac{2}{d}\left(\frac{d}{2M\sigma^2}\right)^{\frac{d}{2}} \frac{1}{\Gamma\left(\frac{d}{2}\right)} x^{\frac{d}{2}}. \tag{9}$$

These two expressions give us another approximation for the pdf and the cdf of the shingle distances for the nearest neighbors, the left-tail of the distribution, that will allow us to calculate the pdf of the smallest distance between the query and the $N$ shingles in the database (Section IV-C).

### B. Fitting $\chi^2$ Distributions

In this section, we use the analysis above of the shingle-distance distribution to fit parameters to observations drawn from two classes of shingles: 1) drawn from pairs of remixed songs, and 2) drawn from pairs of unrelated songs.

As the $\chi^2$ distribution is in the exponential family, we can rewrite the shingle-distance pdf (7) to make it clear what the sufficient statistics are—that is, what numbers we can compute from samples to help us estimate parameters of the distribution. Thus, we put everything in (7) in terms of an exponential

$$\begin{aligned} f_s = \exp\ \Bigg[&\frac{d}{2}\log\left(\frac{d}{2M\sigma^2}\right) - \log\left(\Gamma\left(\frac{d}{2}\right)\right) \\ &+\frac{d-2}{2}\log(x) + \frac{-d}{2M\sigma^2}x\Bigg] \end{aligned} \tag{10}$$

which makes it obvious that the likelihood $\mathcal{L}$ for $N$ data points $x_i$ is

$$\begin{aligned} \mathcal{L} = \exp\ \Bigg[&N\frac{d}{2}\log\left(\frac{d}{2M\sigma^2}\right) - N\log\left(\Gamma\left(\frac{d}{2}\right)\right) \\ &+\frac{d-2}{2}L + \frac{-d}{2M\sigma^2}S\Bigg] \end{aligned} \tag{11}$$

where $S = \sum_i x_i$ and $L = \sum_i \log(x_i)$ are sufficient statistics for this distribution—as the expression for the likelihood depends only on the $\sum$s, and not on how those $\sum$s are made up.

The maximum-likelihood estimates for $d$ and $\sigma^2$ are then given by solving for a stationary point. We now have

[1] Perhaps the upper limit will be reached only in completely atonal music. In simple, harmonic music with a limited chord repertoire, the number of effective degrees of freedom is smaller than $M$.

$$\frac{\partial \log \mathcal{L}}{\partial d} = -\frac{1}{2M\sigma^2}S + \frac{1}{2}L + \frac{N}{2}\log\frac{d}{2M\sigma^2} + \frac{N}{2} - \frac{N}{2}\Psi\left(\frac{d}{2}\right) \quad (12)$$

$$\frac{\partial \log \mathcal{L}}{\partial \sigma^2} = \frac{d}{2M\sigma^4}S - \frac{Nd}{2\sigma^2} \quad (13)$$

where $\Psi$ is the digamma function $(\mathrm{d}\log\Gamma(x)/\mathrm{d}x)(1/\Gamma(x))(\mathrm{d}\Gamma(x)/\mathrm{d}x)$.

By setting (13) to zero, we immediately get

$$\sigma^2 = \frac{1}{M}\frac{S}{N} \quad (14)$$

substituting into 12 we get

$$\frac{\partial \log \mathcal{L}}{\partial d} = \frac{L}{N} + \log\left(\frac{d}{2}\right) - \log\frac{S}{N} - \Psi\left(\frac{d}{2}\right) \quad (15)$$

and solving for zero to find

$$\log\left(\frac{d}{2}\right) - \Psi\left(\frac{d}{2}\right) = \log\frac{S}{N} - \frac{L}{N}. \quad (16)$$

Then defining $Y(x) = \log(x) - \Psi(x)$, and using $Y^{-1}$ to represent the functional inverse of $Y(\cdot)$, we find the maximum-likelihood estimate for the number of independent parameters in the data

$$d = 2Y^{-1}\left(\log\frac{S}{N} - \frac{L}{N}\right). \quad (17)$$

### C. Order Statistics

When doing these retrieval tasks, we find a shingle's nearest neighbors and their corresponding distances. In the "null hypothesis" (unrelated) case, there are a set of $N$ distances, one for each shingle in the database. We wish to find the pdf of these distances, first assuming that the query is unrelated (i.e., independent) to all the other shingles. This is the null hypothesis, and from this, we can estimate a threshold for when two songs are related.

Following the derivation in [16, App. 2], $x_{\min}$ is the minimum value of a set of $N$ values from the distribution $f(x)$ with cdf $F(x)$. We wish to find the probability distribution function of $x_{\min}$. Note that for a value $x$ to be the minimum value, all $N$ values observed must be greater than or equal to that value, which occurs with probability $(1 - F(x))^N$.

Assume the pdf has no weight below a lower limit, which we set without loss of generality to $x = 0$. Further assume that the cumulative distribution for a variable $x$ near the distribution's finite lower limit fits a power law in $x$, so that $F(x) = ax^c$ for $x$ close to 0. We are interested in the distribution of the minimum value $x_{\min}$ from a set of size $N$ of IID values $x_i$ drawn from this distribution.

Consider the following expressions for the cumulative probability of $x_{\min}$, the minimum for a given set of $N$ shingles, which is distributed as $G(x)$ and is an implicit function of $N$

$$G(x) = \Pr(x_{\min} < x) \quad (18)$$

$$= 1 - \Pr(x_{\min} > x) \quad (19)$$

$$= 1 - \Pr(x_i > x, \forall i) \quad (20)$$

$$= 1 - (1 - F(x))^N. \quad (21)$$

In order to derive the distribution of the minimum, let us consider not $x_{\min}$ itself but

$$w = x_{\min}/b_N \quad (22)$$

where we will choose a value for the constant $b_N$ later (as a function of $N$). The cdf $G'(w)$ for the scaled minimum statistic is computed from the cdf of the original data $F(x)$ because for $w$ to be the minimum value of $x$ (scaled by $b_N$), all $N$ values that we have drawn from $x$ must be equal to or greater than that minimum value. This requires that

$$1 - G'(w_0) = P(w > w_0) = [1 - F(wb_N)]^N. \quad (23)$$

With the assumption that the minimum value we are interested in lies in the power-law range (i.e., is sufficiently close to the actual minimum of the distribution for $y$), we replace $F$ in (9) with

$$F(x) = a(x)^c \quad (24)$$

getting

$$1 - G'(w_0) = P(w > w_0) = (1 - a(w_0 b_N)^c)^N. \quad (25)$$

Now, we exploit our freedom to choose $b_N$ (or equivalently, the scale of $w$) to help us. A good choice is

$$b_N = 1/(aN)^{1/c} \quad (26)$$

because this gives us

$$1 - G'(w_0) = (1 - (w_0)^c/N)^N. \quad (27)$$

In the limit of large $N, (1 - z/N)^N$ tends to $e^z$, therefore $1 - G'(w_0) = \exp(-(w_0)^c)$, or by rearranging and dropping the subscript $G'(w) = 1 - \exp(-(w^c))$.

This gives us the cdf as a function of $w$. The pdf is the derivative of this cdf with respect to $w$, which is

$$g'(w) = dG'(w)/dw = cw^{c-1}e^{-(w^c)}. \quad (28)$$

The mean of this density is

$$\bar{w} = \int_0^\infty cw^c \exp(-w^c)\mathrm{d}w \quad (29)$$

and by substituting $u = w^c$ so $\mathrm{d}u = cw^{c-1}\mathrm{d}w$

$$\bar{w} = \int_0^\infty cu^{\frac{1}{c}} \exp(-u)\,\mathrm{d}u = \Gamma\left(1 + \frac{1}{c}\right). \quad (30)$$

Similarly, the expectation of $w^2$ is $\Gamma(1 + (2/c))$.

We expand these expressions for small $x$ with the assumption that $1/c$ and $2/c$ in the above expressions will turn out to be small. Here, $c$ is half the effective number of degrees of freedom $d$, which we hope will be reasonably large for sequences of PCP vectors, but if we estimate a value of $d$ that turns out to be small (say less than about 6 or so) then the following approximation is not valid. By a Taylor series

$$\Gamma(1+x) = \Gamma(1) + \Gamma'(1)x + \frac{1}{2}\Gamma''(1)x^2 \tag{31}$$

$$= 1 - \gamma x + \frac{1}{2}\left(\frac{\pi^2}{6} + \gamma^2\right)x^2 \tag{32}$$

where $\gamma$ is the Euler–Mascheroni constant $\sim 0.577$.

The variance of $w$ is equal to $\sigma_w^2 = \mathbf{E}[w^2] - \mathbf{E}[w]^2$, so, ignoring terms smaller than $(1/c^2)$, is simply $\sigma_w^2 = (\pi^2/6)(1/c^2)$.

### D. Minimum Distances for Shingles

We compute the minimum value of a large set of values, of size $N$, by comparing a shingle to all shingles of an unrelated song. This will be distributed according to (28). Using (22), (26), and then rewriting (9) using the factors in (24) we conclude

$$w = \left[N\frac{2}{d}\left(\frac{d}{2M\sigma^2}\right)^{\frac{d}{2}}\frac{1}{\Gamma\left(\frac{d}{2}\right)}\right]^{\frac{2}{d}} x_{\min}. \tag{33}$$

Therefore, the pdf of the minimum of unrelated shingle distances is

$$g'(w) = \frac{d}{2}w^{\frac{d-2}{2}}e^{-w^{d/2}} \tag{34}$$

which (as discussed above) has mean $\Gamma(1+(2/d)) \approx 1$ and variance $\approx (\pi^2/6)(4/d^2)$.

Given the distribution of minimum distances in the database $x_{\min}$, we can form an estimate of the distribution's parameters. We divide the mean and variance of the pdf in (34) to find

$$\frac{\overline{x_{\min}}^2}{\sigma_{x_{\min}}^2} = \frac{d^2}{4}\frac{6}{\pi^2} \tag{35}$$

which we rearrange to find

$$d^2 = 4\frac{\pi^2}{6}\frac{\overline{x_{\min}}^2}{\sigma_{x_{\min}}^2}. \tag{36}$$

This is an estimate for the number of independent dimensions $d$ given the ratio between the mean and the variance of the minimum distances between a query and all unrelated shingles. This is true for large values of $d$. It is important to note that this estimate is based on the tails of the distribution, while (17) is based on the greater amount of data near the distribution's peak, and thus (17) is probably more accurate in practice.

### E. Related Song Classifier

We derived the distribution in (34) under the assumption that we generated the shingles we are comparing from two distinct, independent processes—that is, the songs from which the shingles are drawn are unrelated. Under this assumption (and all the modeling assumptions above), the distribution for the minimum holds. If an empirical distribution with significant differences is obtained, then we may be able to conclude that one or more of our modeling assumptions is violated.

In the context of related-song detection, the assumption that we expect to see violated is that of the independence of the two songs. If the two songs are related, for example through reuse of content in a remix, then the processes are likely coupled. The signature for this is probably a significantly lower mean value for the minimum.

Most of this modeling aside, the procedure is relatively clear from the point of view of solving a task: compute a baseline "unrelated" distribution for the minimum, approximated using the discussion above with measured mean and variance. Then, the test for "related"ness between two songs is that the mean minimum distance over that song is outside whatever confidence interval we calculate for the mean minimum distance of unrelated songs.

### F. Estimating the Optimal Search Radius

Our motivation for the analysis above was automatic optimization of the search radius $x_{\text{thresh}}$ for retrieval using LSH algorithms. Here, we show how we use the above analysis to achieve this. First, we rearrange (33) to solve for $x_{\min}$ and solve for the factor $w$ using the cdf form of the pdf given in (34), giving

$$G'(w) = \text{cdf}(w) = 1 - e^{-w^{d/2}}. \tag{37}$$

We then perform a reverse lookup on the cdf at the point $\text{cdf}(w) = f_p$ that achieves the required proportion of retrieved nonrelevant shingles (false positives). In our experiments, we set $f_p = 0.01$ yielding an expected proportion of 1% false positives per track. This value is chosen so that the proportion is far below the expected proportion of retrieved shingles for similar tracks. The value of $w$ is then obtained using

$$\ln(w) = \frac{2}{d}\ln(-\ln(1-f_p)). \tag{38}$$

Finally, given $d, \sigma^2$ and $w$, we find $x_{\text{thresh}}$ by rearranging (33)

$$x_{\text{thresh}} = \frac{M\sigma^2 w}{N^{\frac{2}{d}}\frac{d}{2}\left(\frac{2}{d}\right)^{\frac{2}{d}}\left(\frac{1}{\Gamma\left(\frac{d}{2}\right)}\right)^{\frac{2}{d}}}. \tag{39}$$

In our previous work, [10]–[12], we established that LSH algorithms solve music sequence similarity retrieval efficiently, and with the same precision as exact solutions, if the radius is set near the optimum value. The method given in this section provides a solution to the optimum radius given a distribution of distances for nonrelevant data.

In the following section, we present the results of three experiments designed to test the robustness and generality of the solution on a wide range of music-retrieval tasks with different specificities, data sets, and audio features.

## V. Results

We conducted three experiments to test our approach for music similarity, how well our models fit the data, and the accuracy of the optimum-radius solution. The first experiment was fingerprint identification, for which the task was retrieval of the original sources of 49 suspected falsely attributed commercial recordings from a database of 2741 recordings, consisting of 4.5 millions audio shingles spanning 125 h of audio. The second experiment was cover-song retrieval, for which the task was retrieval of the different performances of the work contained in the query performance, the cover song, against a 2741-track database of many different works performed by many different artists. The final experiment was remix retrieval; here, the task was to retrieve remixed versions of 82 query tracks from a database of 2018 tracks consisting of 220 h of audio. Remixes contain a small snippet of the original track's audio, typically a vocal sample, and place it in an entirely new acoustic context so that there is only a small region of similarity between the original and remixed track. These properties make the remix task difficult for automated retrieval.

To solve these tasks with large databases, we need to solve them using a LSH algorithm which requires us to specify hash functions for a given retrieval radius $x_{\text{thresh}}$. The LSH algorithm returns only those shingles that fall with the given radius of the query shingle. To find similar songs, we count the number of collisions between songs. A collision occurs when the distance between pairs of shingles in different tracks falls below the specified minimum distance $x_{\text{thresh}}$. A maximum of one collision is recorded per query shingle for each track in the database. Thus, the maximum collision count is the number of shingles in the query track.

The optimal radius $x_{\text{thresh}}$ is both data and task dependent. We show this by using the same system on the same data to perform two different tasks in Experiments 1 and 2 at different specificities. Similarity for these tasks is entirely defined by the nonsimilar training data, so the methods generalize to any task for which a distance distribution of nonsimilar features can be constructed.

Each of the three tasks—fingerprint, cover song, and remix retrieval—require sensitivity to different musical and acoustic styles. Table I lists the invariance properties required for each of the three tasks. As described below, we chose features for each task that are invariant to the listed properties.

TABLE I
INVARIANCE PROPERTIES FOR THE FINGERPRINT, COVER SONG, AND REMIX TASKS

| Invariance | fingerprint | cover song | Remix |
|---|---|---|---|
| Performer | | X | |
| Environment | | X | X |
| Recording Equip. | | X | |
| Instrumentation | | | X |
| Tempo | | X | X |
| Dynamics | | X | X |
| Volume | X | X | X |
| Pitch/Key | | X | X |
| Harmony | | | X |
| Structure | | X | X |
| Effects | X | | X |
| Audio Encoding | X | | X |

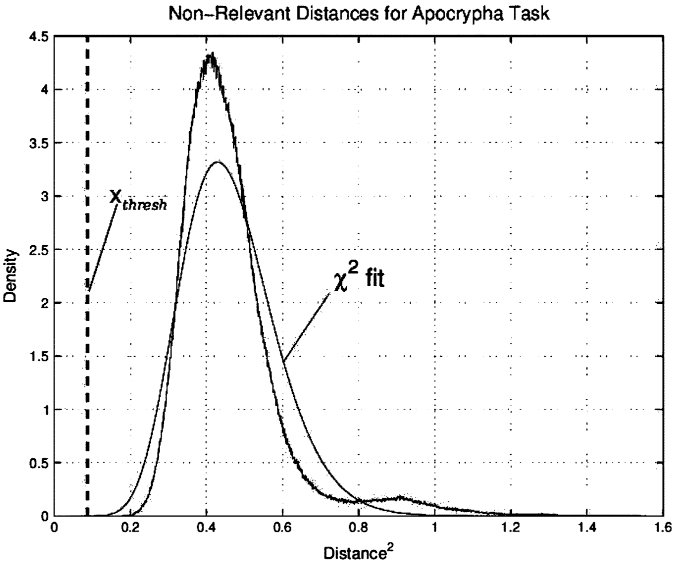

Fig. 3. Distribution of nonrelevant distances for the fingerprint task and their $\chi^2$ fit. The estimated value of $x_{\text{thresh}}$ is 0.095 for the LFCC shingles. This is used as the search radius for near neighbor retrieval algorithms.

## A. Fingerprint

As reported in the classical music press [13], [14], there are 49 commercial recordings of the Chopin Mazurkas by concert pianist Eugen Indjic that were copied, modified, and falsely attributed to pianist Joyce Hatto under the *Concert Artists* record label. Since each of the 49 recordings by Hatto is a modified recording of a different artists' performance, the goal of the first experiment was to robustly identify the misattributed recordings against the complete set of recordings by 125 artists of all 49 Mazurkas, giving a database of 2741 tracks.

The specificity of this task is similar to that of audio fingerprinting. We would like to establish whether two recordings are acoustically identical, but for some degree of signal transformation and distortion such as filtering or time compression/expansion.

*1) Features:* There are between 31 and 62 recordings of each of the Mazurkas in the database, with a mean of 44 recordings for each Mazurka. For each performance, the chord sequences are identical. However, there are variations in the acoustic environment, expressive content such as tempo and dynamics, structure (which repeats are taken and how many times repeated), recording equipment, storage medium, and playback device prior to digitization. Table I shows a list of acoustic and musical invariance properties needed to perform the task.

From Table I, we discern that, for the fingerprint task, our chosen features must be sensitive to the specific details of a performance while being robust to possible effects processing such as filtering, time compress/expand, and artificial reverberation. Given the specificity of the task, we chose LFCC as the features.

*2) Method:* We used (7), (38), and (37) to fit a sample distribution of nonrelevant inter-track shingle distances and obtain an estimate of the minimum distance $x_{\text{thresh}}$ for rejecting samples as nonsimilar. For an exact evaluation of our derived threshold, all distances were computed by exhaustive evaluation using (2). Tracks were ordered using (4), which counts the number of query shingles that are close to any of the track shingles for each track using the indicator function of (3).

Fig. 3 shows the distribution of distances and the $\chi^2$ fit. The estimate of the minimum search radius $x_{\text{thresh}}$ was derived from (38) and (39), using the estimated $d$ and $\sigma^2$ parameters for the $\chi^2$ distribution, estimated cdf factor $z$ and a false alarm rate of

TABLE II
ESTIMATED $\chi^2$ PARAMETERS FOR FINGERPRINT TASK

| Parameter | Value |
|---|---|
| $x_{thresh}$ | 0.095 |
| $d$ | 34.3 |
| $\sigma^2$ | 0.44 |
| $z$ | 0.77 |

1%. Table II shows the values estimated from the nonrelevant data set for each of the parameters.

The estimate of the degrees of freedom value in Table II tells us that out of 360 dimensions in our shingles, 34 of them are independent and determine the $\chi^2$ properties. For uniform signals, over the 30-frame shingle window, the maximum value would be 20, one for each of the LFCC dimensions. A value higher than this informs us that the temporal concatenation of LFCC features is informative for this task.

*3) Retrieval:* We first constructed a suspect query list consisting of known misattributed recordings. The goal of the task was to order the recordings by similarity and for the original track to be at the top of the result list.

The task is difficult because all of the remaining recordings for each work contain the same sequence of notes in the same key, as well as many other consistencies, due to established performance practice over the repertoire. Since, for the 49 Mazurkas, we know which recordings were copied to create the suspect queries, this set was used as a ground-truth. If any of the remaining performances are sufficiently close to the ground-truth, confusion is manifest between the ground-truth and similar tracks given the suspect query.

We counted all shingles in the database that fell on or below the minimum radius $x_{\text{thresh}}$ from the query shingles. This operation can be performed either by visiting all shingles in the database and computing the squared distance between them, or by constructing LSH functions using the minimum-radius estimate. The two methods produce near-identical results for optimal values of $x_{\text{thresh}}$, [11], but the LSH method is several orders of magnitude faster when the radius is set close to the optimal value.

For each query shingle, the distances to all shingles in each database track were measured, and if any fell below the minimum distance $x_{\text{thresh}}$, then the retrieved count was incremented for that track. This was repeated for all 2741 tracks in the database. The tracks were ordered by the number of matched query shingles, thus constituting a one-to-many mapping between query shingles and track shingles.

*4) Fingerprint Results:* The results of this first experiment were surprising. All 49 misattributed recordings were retrieved first in the ordered result list, using the radius-search method and the estimated $x_{\text{thresh}}$. In the first instance, we retrieved the correct recording for each suspect recording from the remaining performances of the same work only. We then repeated the experiment using the entire collection of 2741 recordings as interference for each retrieval trial, and the result remained perfect. We also tried repeating the experiment using a different method of ordering, the average distance of the ten nearest returned shingles [11], [12]; with this method of ordering, the score decreased. Some of the target recordings were retrieved further down the result list than for our proposed retrieved-shingle-count method.

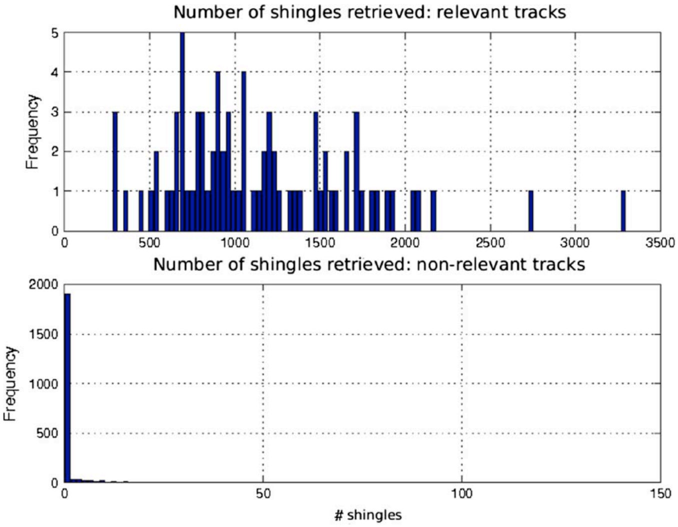

Fig. 4. Histogram of retrieved shingle counts for the fingerprint task. The $x$ axis indicates how many query shingles matched the database shingles for relevant and nonrelevant tracks, and the $y$ axis shows how often this level of similarity occurs. The upper graph shows the counts for relevant data and the lower shows counts for nonrelevant data. Note that the scales are different on the two graphs. Performance in the task is related to the degree of separation between these graphs.

Fig. 4 shows the distribution of retrieved-shingle counts for relevant and nonrelevant tracks, respectively. In this figure, we can see that the two distributions do not overlap; hence, the perfect result is due to the high degree of separation between the number of retrieved shingles, between the relevant and non-relevant classes. The comparison with the 10-nearest-neighbor ordering method suggests that the task is not trivial and that our chosen method is robust for this task.

## *B. Experiment 2: Cover Song Retrieval*

Our second experiment used the same audio data as Experiment 1, but we made the task substantially harder by decreasing the specificity. The cover-song retrieval task is defined as follows: given a query performance of one of the Chopin Mazurkas, retrieve all the performances of the same work given the entire 2741 track database. The task is difficult because performances are by different artists, with different expressive interpretations. Furthermore, each performance has different structuring, due to choices over performing repeats and how many times to play them. The required invariance properties for this task are listed in Table I, and we used the PCP features for this test.

*1) Method:* The method for retrieval was the same as Experiment 1. We formed a database of audio shingles consisting of all 2741 performances of the Chopin Mazurkas by 125 different artists. Given a single performance of each Mazurka, the goal was to retrieve the remaining performances of the same cover song.

To fit the nonrelevant distance distribution for the task, we collected a sample of 10-s segments drawn from 40 nonrelated performances, i.e., different versions of the song performed by different artists. The squared distances of these samples were

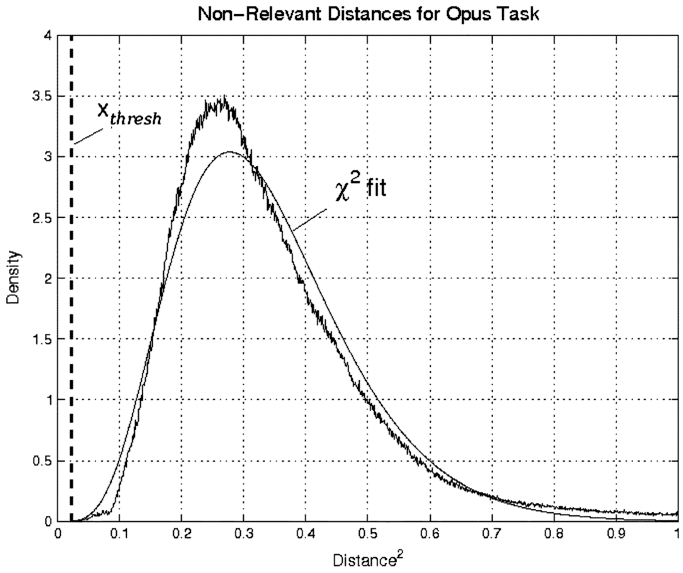

Fig. 5. Distribution of distances for the cover song task and corresponding $\chi^2$ fit. The estimation for the minimum distance yielded $x_{\mathrm{thresh}} = 0.01$ for the given background data using smoothed 12-chroma PCP shingles.

TABLE III
ESTIMATED $\chi^2$ PARAMETERS FOR COVER SONG TASK

| Parameter | Value |
|---|---|
| $x_{thresh}$ | 0.011 |
| $d$ | 8.35 |
| $\sigma^2$ | 0.32 |
| $z$ | 0.33 |

computed, rejecting intra-track distances, and we fit the distance distribution using the $\chi^2$ estimation methods. Fig. 5 shows the distribution of nonrelevant shingle distances for the task, as well as the $\chi^2$ fit and $x_{\mathrm{thresh}}$ estimate. Table III shows the parameters estimated from the nonrelevant distance distribution for the cover-song task. The false-alarm rate was set to 1%, as in the previous task, so the only differences in the method were the features used and the set of nonrelevant data used to fit the distribution.

The degrees-of-freedom estimate for the data in this task is substantially lower than for the fingerprint task. This makes sense because we have designed the features to be more tolerant of differences between recordings, thus reducing the amount of variability between them.

*2) Cover Song Results:* The precision-recall graph for the cover-song-retrieval task are shown in Fig. 6. Overall, the precision was very high for recall rates below 90%. For most of the 49 Mazurkas, there were two to three outliers in our database. On inspection, these were typically early recordings that were transferred from 78 RPM shallac discs and therefore contained surface noise and spectral distortion due to the early period recording process. Additionally, the cutoff frequency for these tracks was typically much lower than the remaining tracks. These results suggest that a near-perfect score can be obtained for cover song retrieval if outlying recordings are first removed or preprocessed to make them compatible with the system.

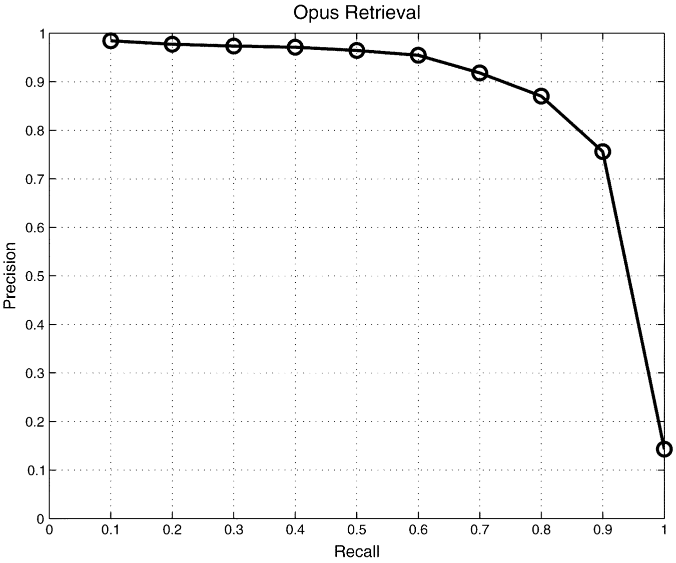

Fig. 6. Precision-recall evaluation plot for the cover song retrieval task.

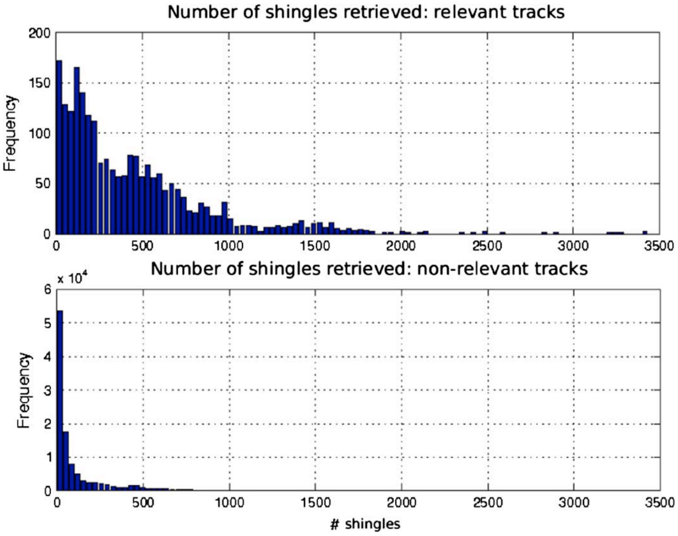

Fig. 7. Distribution of retrieved shingle counts for the Opus. See Fig. 4 for more details.

Fig. 7 shows the distribution of the number of retrieved shingles over all query tracks for relevant and nonrelevant tracks. The precision-recall graph suggests that the distributions of counts do not overlap on a per-query basis. In other words, when a query track produces a relatively high number of hits in nonrelevant tracks, the relevant tracks get a proportionally higher number of hits, thereby preserving the correct ordering. This makes the count method a robust classifier.

We computed the false positive rate for retrieved minimum-distance shingles. The total number of query shingles was 2209 of which 2129 minimum distance shingles were retrieved from relevant tracks in the database and 80 were retrieved from nonrelevant tracks. Therefore, the experiment yielded a false positive rate of 3.62%. This rate compares favorably with the 1% false positive rate used to estimate the radius threshold for retrieved shingles.

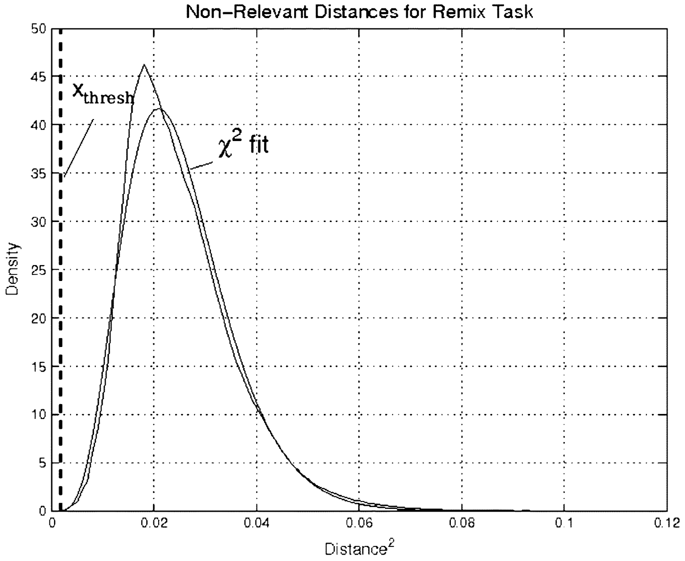

Fig. 8. Distribution of distances for the Remix task and $\chi^2$ fit. The estimation for the minimum distance yielded $x_{\text{thresh}} = 0.004$ for a false alarm rate of 1%.

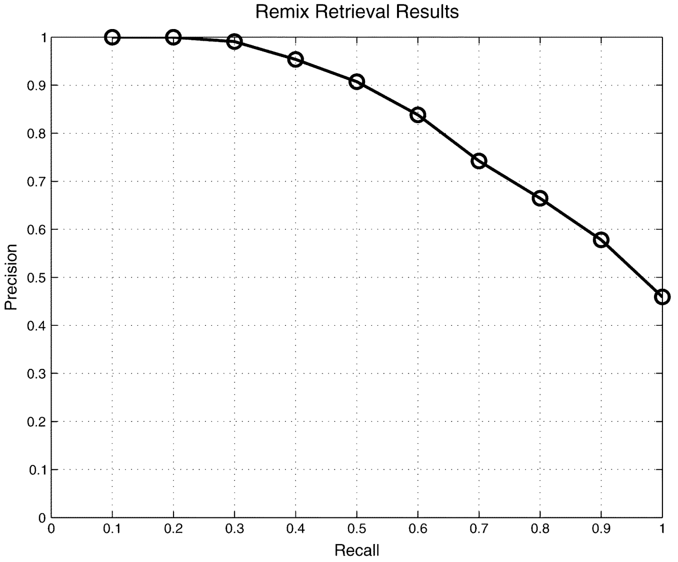

Fig. 9. Precision-recall evaluation of the Remix task.

TABLE IV
ESTIMATED $\chi^2$ PARAMETERS FOR REMIX TASK

| Parameter | Value |
|---|---|
| $x_{thresh}$ | 0.004 |
| $d$ | 7.15 |
| $\sigma^2$ | 0.024 |
| $z$ | 2.08 |

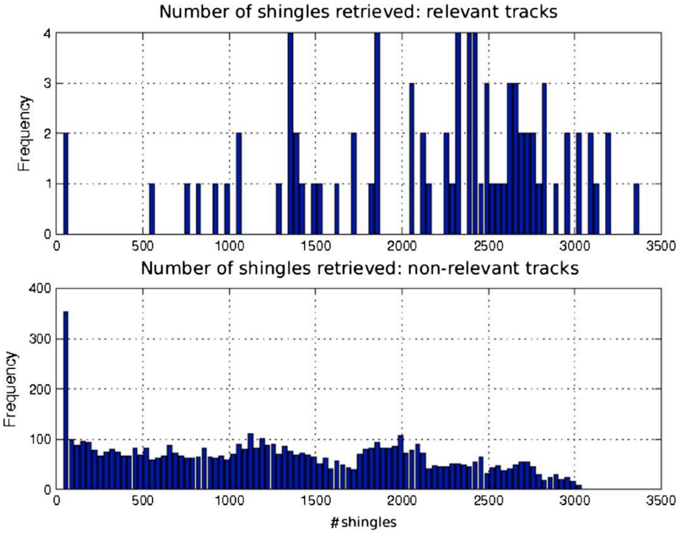

Fig. 10. Distribution of retrieved shingle counts for the remix task. See Fig. 4 for more details.

### C. *Experiment 3: Remix Retrieval*

To test the method on a different data set, consisting of pop music and jazz recordings, we devised a third experiment with specificities falling between Experiments 1 and 2. The goal in this task is to retrieve remixed versions of a query track from the database of 2018 recordings consisting of popular and jazz recordings [12].

This task required a high degree of robustness to many different acoustic and musical properties. To meet the invariance criteria, we used 12-dimensional PCP features that were not smoothed because we wanted to retain sensitivity to any specific audio content from an original track that was used for a remix.

*1) Method:* A background of nonrelevant distances was sampled from 40 unrelated tracks, 10 s of shingles from each, forming a total of 400 s of distance data. The distances between all the background vectors were calculated using convolution, which implements Euclidean distance in the case of unit vectors.

The minimum-radius threshold was computed using (38) and (39), see Fig. 8. The estimated parameters are shown in Table IV for the distribution of nonrelevant distances.

A set of queries was selected consisting of 82 tracks with three to ten remixes each; this formed the ground-truth set. These 82 tracks included all the remixes, such that the base set was 20 tracks with remix versions expanding this set to 82. A further 2000 tracks, both similar and contrasting in genres, were added as interference, including 226 tracks by similar artists as the 82 remix tracks. Each query shingle was compared against the shingles in each database track. If the query shingle resulted in a match, with distance below $x_{\text{thresh}} = 0.004$, then the count of minimum-value matches for the track was incremented. The counts were used to order the tracks with the track having the highest minimum-value counts placed first.

*2) Results:* Fig. 9 shows the precision-recall graph for the remix retrieval task. The performance is satisfactory given the difficulty of the task, with precision at 50% for 100% recall and substantially higher for lower recall rates. The break-even point in precision-recall for this task was somewhere around 75% for the given features. Setting the search radius to a higher value did not improve the result and substantially slows down retrieval when using LSH.

Fig. 10 shows the distribution of minimum-distance counts for the relevant and nonrelevant tracks, respectively. The top graph shows the distribution of the number of relevant shingles retrieved over all remixed tracks. The bottom graph shows the distribution of number of shingles retrieved for all nonrelevant tracks. The distributions of counts overlap substantially more than the previous two experiments indicating that the task is more difficult.

### *D. Discussion*

The results from our three experiments demonstrate that we were able to correctly order the tracks by relevance to the queries by rejecting the hypothesis that they were nonrelevant. This method has two main advantages over other methods such as those based on averaging the retrieved distances per track. First, hypothesis testing improves performance without requiring extensive empirical evaluation of parameters, such as $k$ in the $k$-NN method. Second, and more importantly for our purposes, the minimum value, $x_{\mathrm{thresh}}$ is a distance which is interpreted as an absolute threshold on the distances of shingles to retrieve. This property means that it can be used to estimate the threshold for implementing our retrieval algorithm using LSH, which we have shown in our previous work to perform with the same accuracy as the exact methods employed here, but more efficiently—approximately two orders of magnitude faster to retrieve relevant tracks from a 4.5 million shingle database.

Furthermore, the results indicate that our analysis of distance distributions is both correct and appropriate to the tasks we set out to solve. We believe that this analysis will lead to greater understanding of how retrieval systems perform for music tasks and may lead to new and improved algorithms for retrieval.

## VI. Conclusion

This paper describes the statistical basis of nearest neighbors in a music-retrieval task. We assume that high-dimensional features are distributed as Gaussian random variables and derived expressions for the distance to a query's nearest neighbors. This allows us to derive analytical expressions for the correct threshold to use when deciding whether the nearest neighbor is musically similar to the original query. We derive an algorithm that determines the statistical properties of real data, and we use this to estimate the degree of independence in real data. While our problem is motivated with a musical example, the analysis is general and applies to any similar high-dimensional database which can be modeled with Gaussian statistics.

We show that a similarity measure between songs using only small parts of the song can be used effectively to identify songs that are related as remixes. We use the distributions of inter-song shingle distances and show that separation of the two distributions can be achieved by choosing a suitable threshold on the distances, and that this threshold could be estimated from examples using a $\chi^2$ distributions. With a suitable kernel space, we show that a threshold classifier can be used for robust audio matching for mid-specificity problems such as fingerprint, cover song, and remix recognition.

We evaluate our analysis and demonstrate that we can apply the results to real data using three tasks of varying specificity. The fingerprint task allows us to find recordings that are attributed to the wrong artist—the most specific question. The cover-song task requires that we find different recordings of the same musical piece. Finally, in the remix task, we find recordings based on the same musical piece, but rearranged in some manner to make them different. In each case, we describe the multidimensional feature vector, analyze its statistical properties and determine a true measure of its independence, derive a threshold for our decision process, and then show excellent results on the task.

In future work, we hope to use more robust features to see if the degree of separation between distributions can be improved; this will lead to increased performance and a greater degree of generalization of our results.

## References

[1] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Commun. ACM*, vol. 51, no. 1, pp. 117–122, 2008.

[2] S. Baluja and M. Covell, "Audio fingerprinting: Combining computer vision & data stream processing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 15–20, 2007, vol. 2, pp. II-213–II-216.

[3] S. Baluja and M. Covell, "Learning forgiving hash functions: Algorithms and large scale tests," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 2663–2669.

[4] M. A. Bartsch and G. H. Wakefield, "To catch a chorus: Using chroma-based representations for audio thumbnailing," in *Proc. WASPAA*, 2001, pp. 15–18.

[5] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'nearest neighbor' meaningful?," in *ICDT'99: Proc. 7th Int. Conf. Database Theory*, London, U.K., 1999, pp. 217–235.

[6] C. Böhm, "A cost model for query processing in high dimensional data spaces," *ACM Trans. Database Syst.*, vol. 25, no. 2, pp. 129–178, 2000.

[7] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the web," in *Proc. WWW6'97*, Apr. 1997, pp. 391–404, Elsevier Science.

[8] C. J. 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.

[9] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of algorithms for audio fingerprinting," in *Proc. Int. Workshop Multimedia Signal Process.*, Dec. 2002, U.S. Virgin Islands.

[10] M. Casey and M. Slaney, "The importance of sequences in music similarity," in *Proc. ICASSP*, 2006, pp. V-5–V-8.

[11] M. Casey and M. Slaney, "Song intersection by approximate nearest neighbor search," in *Proc. ISMIR*, 2006, pp. 144–149.

[12] M. Casey and M. Slaney, "Fast Recognition of Remixed Music Audio," in *Proc. ICASSP*, Honolulu, HI, May 2007, pp. 1425–1428.

[13] N. Cook and C. Sapp, "Purely coincidental? Joyce Hatto and Chopin's Mazurkas," Royal Holloway, Univ. of London, London, U.K., 2007 [Online]. Available: http://www.charm.rhul.ac.uk/content/contact/hatto_article.html

[14] "Revenge of the Fraudster Pianist," The Daily Mail, U.K., Feb. 24, 2007.

[15] T. Darrell, P. Indyk, and G. Shakhnarovich, Eds., *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*. Cambridge, U.K.: MIT Press, 2005.

[16] D. R. Cox and D. V. Hinkley, *Theoretical Statistics*. London, U.K.: Chapman & Hall, 1974.

[17] J. Herre, E. Allamanche, O. Hellmuth, and T. Kastner, "Robust identification/fingerprinting of audio signals using spectral flatness features," *J. Acoust. Soc. Amer.*, vol. 111, no. 5, pp. 2417–2417, 2002.

[18] J. Haitsma, T. Kalker, and J. Oostveen, "A highly robust fingerprinting system," in *Proc. ISMIR*, Paris, France, 2002, pp. 107–115.

[19] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. ISMIR*, Paris, 2002, pp. 107–115.

[20] P. Indyk and R. Motwani, "Approximate nearest neighbor: Towards removing the curse of dimensionality," in *Proc. Symp. Theory Comput.*, 1998, pp. 604–613.

[21] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, Jun. 2005, vol. 1, pp. 597–604.

[22] M. Miller, M. Rodriguez, and I. Cox, "Audio fingerprinting: Nearest neighbor search in high dimensional binary spaces," in *Proc. IEEE Workshop Multimedia Signal Process.*, 2002, pp. 182–185.

[23] M. Mueller, F. Kurth, and M. Clausen, "Audio matching via chroma-based statistical features," in *Proc. ISMIR*, London, U.K., Sep. 2005, pp. 288–295.

[24] E. Pampalk, A. Flexer, and G. Widmer, "Improvements of audio-based music similarity and genre classificaton," in *Proc. ISMIR*, 2005, pp. 628–633.

[25] U. Shaft and R. Ramakrishnan, "Theory of nearest neighbors indexability," *ACM Trans. Database Syst.*, vol. 31, no. 3, pp. 814–838, 2006.

[26] M. Slaney and M. Casey, "Locality-sensitive hashing for finding nearest neighbors," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 128–131, Mar. 2008.

[27] Y. Tao, J. Zhang, D. Papadias, and N. Mamoulis, "An efficient cost model for optimization of nearest neighbor search in low and medium dimensional spaces," *IEEE Trans. Knowledge Data Eng.*, vol. 16, no. 10, pp. 1169–1184, Oct. 2004.

[28] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002.

[29] A. L. Wang and J. O. Smith III, "System and Methods for Recognizing Sound and Music Signals in High Noise and Distortion," United States Patent 6990453, 2006.

[30] E. Weinstein and P. Moreno, "Music identification with weighted finite-state transducers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 15–20, 2007, vol. 2, pp. II-689–II-692, no.,.

[31] C. Yang, "Efficient acoustic index for music retrieval with various degrees of similarity," *Proc. ACM Multimedia*, pp. 584–591, 2002.

**Michael Casey** (M'99) received the Ph.D. degree from the MIT Media Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, in 1998

He is a Professor of computer science at Goldsmiths College, University of London, U.K., and a Professor of music at Dartmouth College, Hanover, NH. After the Ph.D. degree, he was a Research Scientist at Mitsubishi Electric Research Laboratories (MERL), Cambridge, until joining Goldsmiths in 2004. He is the founding Principal Investigator at Goldsmiths for the Online Music Recognition and Searching II (OMRAS2) project funded by the U.K. EPSRC.

**Christophe Rhodes** received the B.S. degree in physics in 2000 and the Ph.D. degree in applied mathematics in 2004, both from the University of Cambridge, U.K.

He is a Lecturer in the Intelligent Sound and Music Systems Group, Computer Science Department, Goldsmiths College, University of London, U.K. His current research interests span musical information retrieval and representation, user interfaces, and language design and implementation.

**Malcolm Slaney** (SM'01) received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN.

He is a Researcher with Yahoo! Research, Sunnyvale, CA, and a Consulting Professor at Stanford University, Stanford, CA. He is a coauthor of the book *Principles of Computerized Tomographic Imaging* (a Classic in Applied Mathematics by the Society for Industrial and Applied Mathematics, 2001) and coeditor of the book *Computational Models of Hearing* (IOS Press, 2001).