A Review of Algorithms for Audio Fingerprinting

Pedro Cano and Eloi Batlle
Universitat Pompeu Fabra
Barcelona, Spain
Email: {pedro.cano, eloi.batlle}@iua.upf.es

Ton Kalker and Jaap Haitsma
Philips Research Eindhoven
Eindhoven, The Netherlands
Email: ton.kalker@ieee.org, jaap.haitsma@philips.com

***Abstract*— An audio fingerprint is a content-based compact signature that summarizes an audio recording. Audio Fingerprinting technologies have recently attracted attention since they allow the monitoring of audio independently of its format and without the need of meta-data or watermark embedding. The different approaches to fingerprinting are usually described with different rationales and terminology depending on the background: Pattern matching, Multimedia (Music) Information Retrieval or Cryptography (Robust Hashing). In this paper, we review different techniques mapping functional parts to blocks of a unified framework.**

I. Introduction

Audio fingerprinting is best known for its ability to link unlabeled audio to corresponding metadata (e.g. artist and song name), regardless of the audio format. Although there are more applications to audio fingerprinting, such us: Content-based integrity verification or watermarking support, this review focuses primarily on identification. Audio fingerprinting or Content-based audio identification (CBID) systems extract a perceptual digest of a piece of audio content, i.e. the fingerprint and store it in a database. When presented with unlabeled audio, its fingerprint is calculated and matched against those stored in the database. Using fingerprints and matching algorithms, distorted versions of a recording can be identified as the same audio content.

A source of difficulty when automatically identifying audio content derives from its high dimensionality and the significant variance of the audio data for perceptually similar content. The simplest approach that one may think of – the direct comparison of the digitalized waveform – is neither efficient not effective. An efficient implementation of this approach could use a hash method, such as MD5 (Message Digest 5) or CRC (Cyclic Redundancy Checking), to obtain a compact representation of the binary file. In this setup, one compares the hash values instead of the whole files. However, hash values are fragile, a single bit flip is sufficient for the hash to completely change. Of course this setup is not robust to compression or minimal distorions of any kind and, in fact, it cannot be considered as content-based identification since it does not consider the content, understood as information, just the bits.

An ideal fingerprinting system should fulfill several requirements. It should be able to accurately identify an item, regardless of the level of compression and distortion or interference in the transmission channel. Depending on the application, it should be able to identify whole titles from excerpts a few seconds long (property known as granularity or robustness to cropping), which requires methods for dealing with shifting, that is lack of synchronization between the extracted fingerprint and those stored in the database. It should also be able to deal with other sources of degradation such as pitching (playing audio faster or slower), equalization, background noise, D/A-A/D conversion, speech and audio coders (such as GSM or MP3), etc. The fingerprinting system should also be computationally efficient. This is related to the size of the fingerprints, the complexity of the search algorithm and the complexity of the fingerprint extraction.

The design principles behind audio fingerprinting are recurrent in several research areas. Compact signatures that represent complex multimedia objects are employed in Information Retrieval for fast indexing and retrieval. In order to index complex multimedia objects it is necessary to reduce their dimensionality (to avoid the "curse of dimensionality") and perform the indexing and searching in the reduced space [1]–[3]. In analogy to the cryptographic hash value, content-based digital signatures can be seen as evolved versions of hash values that are robust to content-preserving transformations [4], [5]. Also from a pattern matching point of view, the idea of extracting the essence of a class of objects retaining the main its characteristics is at the heart of any classification system [6]–[10].

II. General Framework

In spite of the different rationales behind the identification task, methods share certain aspects. As depicted in Fig.1, there are two fundamental processes: the fingerprint extraction and the matching algorithm. The fingerprint extraction derives a set of relevant perceptual characteristics of a recording in a concise and robust form. The fingerprint requirements include:

- Discrimination power over huge numbers of other fingerprints,
- Invariance to distortions,
- Compactness,
- Computational simplicity.

The solutions proposed to fulfill the above requirements imply a trade-off between dimensionality reduction and information loss. The fingerprint extraction consists of a front-end and a fingerprint modeling block (see Fig.2). The front-end computes a set of measurements from the signal (see Section III). The fingerprint model block defines the final fingerprint representation, e.g: a vector, a trace of vectors, a codebook,

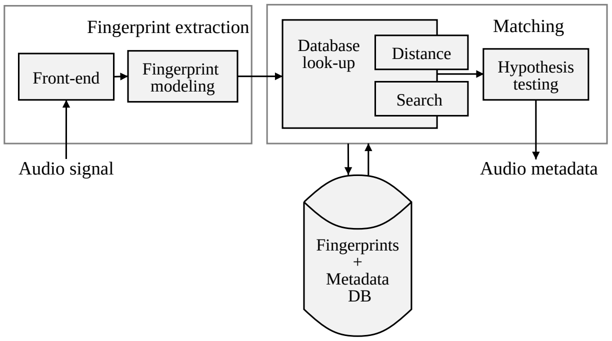

Fig. 1. Content-based Audio Identification Framework.

a sequence of indexes to HMM sound classes, a sequence of error correcting words or musically meaningful high-level attributes (see Section IV).

Given a fingerprint derived from a recording, the matching algorithm searches a database of fingerprints to find the best match. A way of comparing fingeprints, that is a distance, is therefore needed (see Section V-A). Since the number of comparison is high and the distance can be expensive to compute, we require methods that speed up the search. It is common to see methods that use a simpler distance to quickly discard candidates and the more correct but expensive distance for the reduced set of candidates. There are also methods that pre-compute some distances off-line and build a data structure that allows reducing the number of computations to do on-line (see Section V-B). According to [1], good searching methods should be :

- Fast: Sequential scanning and distance calculation can be too slow for huge databases.
- Correct: Should return the qualifying objects, without missing any – low False Rejection Rate (FRR).
- Memory efficient: They should require small space overhead.
- Easily updatable: They should allow to easily insert, delete and update objects.

The last block of the system – the hypothesis testing (see Fig.1) – computes a reliability measure indicating how sure the system is about an identification (see Section VI).

III. Front-End

The front-end converts an audio signal into a sequence of relevant features to feed the fingerprint model block (see Fig.2). Several driving forces co-exist in the design of the front-end:

- Dimensionality reduction
- Perceptually meaningful parameters (similar to those used by the human auditory system)
- Invariance or robustness (to channel distortions, background noise, etc.)
- Temporal correlation (systems that capture spectral dynamics).

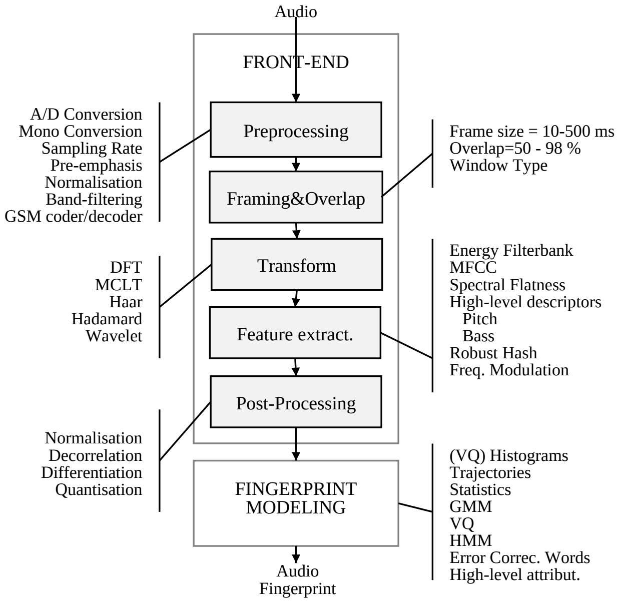

Fig. 2. Fingerprint Extraction Framework: Front-end (top) and Fingerprint modeling (bottom).

In some applications, where the audio to identify is coded, for instance in mp3, it is possible to by-pass some blocks and extract the features from the audio coded representation.

A. *Preprocessing*

In a first step, the audio is digitalized (if necessary) and converted to a general format: Often to raw format (16 bits PCM), to mono averaging left and right channels, to a certain sampling rate (ranging from 5 to 44.1 KHz). Sometimes the audio is preprocessed to simulate the channel, e.g: band-pass filtered in a telephone identification task. Other types of processing are a GSM coder/decoder in a mobile phone identification system, pre-emphasis, amplitude normalization (bounding the dynamic range to (-1,1)).

B. *Framing&Overlap*

A key assumption in the measurement of characteristics is that the signal can be regarded as stationary over an interval of a few milliseconds. Therefore, the signal is divided into frames of a size comparable to the variation velocity of the underlying acoustic events. The number of frames computed per second is called frame rate. A tapered window function is applied to each block to minimize the discontinuities at the beginning and end. Overlap must be applied to assure robustness to shifting (i.e. when the input data is not perfectly aligned). There is again a trade-off when choosing the above values between the rate of change in the spectrum and system complexity.

C. Linear Transforms: Spectral Estimates

The idea behind linear transforms is the transformation of the set of measurements to a new set of features. If the transform is suitably chosen, the redundancy is significantly reduced. There are optimal transforms in the sense of information packing and decorrelation properties, like Karhunen-Loève (KL) or Singular Value Decomposition (SVD) [9]. These transforms, however, are problem dependent and computationally complex. For that reason, lower complexity transforms using fixed basis vectors are common. Most CBID methods therefore use standard transforms from time to frequency domain to facilitate efficient compression, noise removal and subsequent processing. Lourens [11], (for computational simplicity), and Kurth *et al.* [12], (to model highly distorted sequences, where the time-frequency analysis exhibits distortions), use power measures. The power can still be seen as a simplified time-frequency distribution, with only one frequency bin.

The most common transformation is the Fast Fourier Transform (FFT). Some other transforms have been proposed: the Discrete Cosine Transform (DCT), the Haar Transform or the Walsh-Hadamard Transform [2]. Richly *et al.* did a comparison of the DFT and the Walsh-Hadamard Transform that revealed that the DFT is generally less sensitive to shifting [13]. The Modulated Complex Transform (MCLT) used by Mihçak *et al.* [5] and also by Burges *et al.* [14] exhibits approximate shift invariance properties [5].

D. Feature Extraction

Once on a time-frequency representation, additional transformations are applied in order to generate the final acoustic vectors. In this step, we find a great diversity of algorithms. The objective is again to reduce the dimensionality and, at the same time, to increase the invariance to distortions. It is very common to include knowledge of the transduction stages of the human auditory system to extract more perceptually meaningful parameters. Therefore, many systems extract several features performing a critical-band analysis of the spectrum (see Fig.3). In [6], [15], Mel-Frequency Cepstrum Coefficients (MFCC) are used. In [7], the choice is the Spectral Flatness Measure (SFM), which is an estimation of the tone-like or noise-like quality for a band in the spectrum. Papaodysseus *et al.* [16] presented the "band representative vectors", which are an ordered list of indexes of bands with prominent tones (i.e. with peaks with significant amplitude). Energy of each band is used by Kimura *et al.* [3]. Haitsma *et al.* use the energies of 33 bark-scaled bands to obtain their "hash string", which is the sign of the energy band differences (both in the time and the frequency axis) [4].

Sukittanon and Atlas claim that spectral estimates and related features only are inadequate when audio channel distortion occurs [8]. They propose modulation frequency analysis to characterize the time-varying behavior of audio signals. In this case, features correspond to the geometric mean of the modulation frequency estimation of the energy of 19 bark-spaced band-filters.

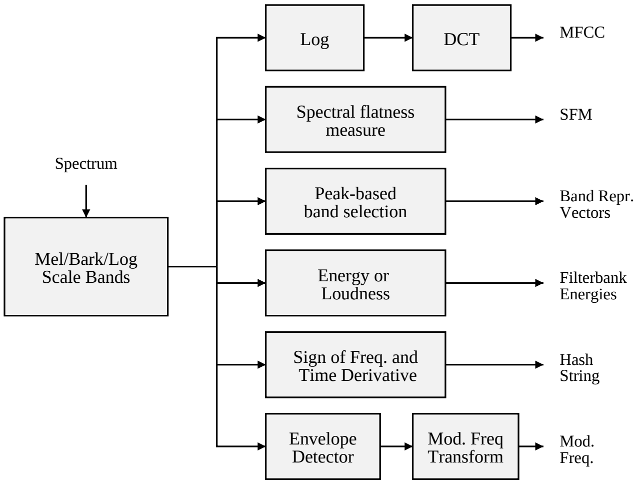

Fig. 3. Feature Extraction Examples

Approaches from music information retrieval include features that have proved valid for comparing sounds: harmonicity, bandwidth, loudness [15], ZCR, etc.

Burges *et al.* point out that the features commonly used are heuristic, and as such, may not be optimal [14]. For that reason, they use a modified Karhunen-Love transform, the Oriented Principal Component Analysis (OPCA), to find the optimal features in an "unsupervised" way. If PCA (KL) finds a set of orthogonal directions which maximize the signal variance, OPCA obtains a set of possible non-orthogonal directions which take some predefined distortions into account.

E. Post-processing

Most of the features described so far are absolute measurements. In order to better characterize temporal variations in the signal, higher order time derivatives are added to the signal model. In [6] and [17], the feature vector is the concatenation of MFCCs, their derivative (delta) and the acceleration (delta-delta), as well as the delta and delta-delta of the energy. Some systems only use the derivative of the features, not the absolute features [7], [12]. Using the derivative of the signal measurements tends to amplify noise [10] but, at the same time, filters the distortions produced in linear time invariant, or slowly varying channels (like an equalization). Cepstrum Mean Normalization (CMN) is used to reduce linear slowly varying channel distortions in [17]. If Euclidean distance is used (see Section V-A), mean subtraction and component wise variance normalization are advisable. Some systems compact the feature vector representation using transforms (e.g: PCA [6], [17]).

It is quite common to apply a very low resolution quantization to the features: ternary [13] or binary [4], [12]. The purpose of quantization is to gain robustness against distortions [4], [12], normalize [13], ease hardware implementations, reduce the memory requirements and for convenience in subsequent parts of the system. Binary sequences are required to extract error correcting words utilized in [5], [12]. In [5], the discretization is designed to increase randomness in order to minimize fingerprint collision probability.

IV. Fingerprint Models

The fingerprint modeling block usually receives a sequence of feature vectors calculated on a frame by frame basis. Exploiting redundancies in the frame time vicinity, inside a recording and across the whole database, is useful to further reduce the fingerprint size. The type of model chosen conditions the distance metric and also the design of indexing algorithms for fast retrieval (see Section V).

A very concise form of fingerprint is achieved by summarizing the multidimensional vector sequences of a whole song (or a fragment of it) in a single vector. Etantrum [18] calculates the vector out of the means and variances of the 16 bank-filtered energies corresponding to 30 sec of audio ending up with a signature of 512 bits. The signature along with information on the original audio format is sent to a server for identification. Musicbrainz' TRM signature [19] includes in a vector: the average zero crossing rate, the estimated beats per minute (BPM), an average spectrum and some more features to represent a piece of audio (corresponding to 26 sec). The two examples above are computationally efficient and produce a very compact fingerprint. They have been designed for applications like linking mp3 files to metadata (title, artist, etc.) and are more tuned for low complexity (both on the client and the server side) than for robustness (cropping or broadcast streaming audio).

Fingerprints can also be sequences (traces, trajectories) of features. This fingerprint representation is found in [15], and also in [4] as binary vector sequences. The fingerprint in [16], which consists on a sequence of "band representative vectors", is binary encoded for memory efficiency.

Some systems, include high-level musically meaningful attributes, like rhythm (BPM) or prominent pitch (see [19] and [15]).

Following the reasoning on the possible sub-optimality of heuristic features, Burges *et al.* [14] employ several layers of OPCA to decrease the local statistical redundancy of feature vectors with respect to time. Besides reducing dimensionality, extra robustness requisites to shifting and pitching are accounted in the transformation.

"Global redundancies" within a song are exploited in [7]. If we assume that the features of a given audio item are similar among them, a compact representation can be generated by clustering the feature vectors. The sequence of vectors is thus approximated by a much lower number of representative code vectors, a codebook. The temporal evolution of audio is lost with this approximation. Also in [7], short-time statistics are collected over regions of time. This results in both higher recognition, since some temporal dependencies are taken into account, and a faster matching, since the length of each sequence is also reduced.

[6] and [17] use a fingerprint model that further exploits global redundancy. The rationale is very much inspired on speech research. In speech, an alphabet of sound classes, i.e. phones can be used to segment a collection of raw speech data into text achieving a great redundancy reduction without "much" information loss. Similarly, we can view a corpus of music, as sentences constructed concatenating sound classes of a finite alphabet. "Perceptually equivalent" drum sounds, for instance, occur in a great number of pop songs. This approximation yields a fingerprint which consists in sequences of indexes to a set of sound classes representative of a collection of audio items. The sound classes are estimated via unsupervised clustering and modeled with Hidden Markov Models (HMMs). Statistical modeling of the signal's time course allows local redundancy reduction. The fingerprint representation as sequences of indexes to the sound classes retains the information on the evolution of audio through time.

In [5], discrete sequences are mapped to a dictionary of error correcting words. In [12], the error correcting codes are at the basis of their indexing method.

V. distances and searching methods

A. Distances

Distance metrics are very much related to the type of model chosen. When comparing vector sequences, a correlation is common. The Euclidean distance, or slightly modified versions that deal sequences of different lengths, are used for instance in [15]. In [8], the classification is Nearest Neighbor using a cross entropy estimation. In the systems where the vector feature sequences are quantized, a Manhattan distance (or Hamming when the quantization is binary) is common [4], [13]. Mihçak *et al.* [5] suggest that another error metric, which they call "Exponential Pseudo Norm" (EPN), could be more appropriate to better distinguish between close and distant values with an emphasis stronger than linear.

So far we have presented an identification framework that follows a template matching paradigm [9]: both the reference patterns – the fingerprints stored in the database – and the test pattern – the fingerprint extracted from the unknown audio – are in the same format and are compared according to some distance metric, e.g: hamming distance, a correlation and son on. In some systems, only the reference items are actually "fingerprints" – compactly modeled as a codebook or a sequence of indexes to HMMs [7], [17]. In these cases, the distances are computed directly between the feature sequence extracted from the unknown audio and the reference audio fingerprints stored in the repository. In [7], the feature vector sequence is matched to the different codebooks using a distance metric. For each codebook, the errors are accumulated. The unknown item is assigned to the class which yields the lowest accumulated error. In [17], the feature sequence is run against the fingerprints (a concatenation of indexes pointing at HMM sound classes) using the Viterbi algorithm. The most likely passage in the database is selected.

B. Searching methods

Besides the definition of a distance metric for fingerprint comparison, a fundamental issue for the usability of a system is how to efficiently do the comparisons of the unknown audio against the possibly million fingerprints. The method depends on the fingerprint represention. Vector spaces allow the use of efficient existing spatial access methods [1]. The general goal is to build a data structure, an index, to reduce the number of distance evaluations when a query is presented. As stated by Chávez *et al.*, most indexing algorithms for proximity searching build sets of equivalence classes, discard some classes and search exhaustively the rest [20] (see for example [3]). The idea of using a simpler distance to quickly eliminate many hypothesis and the use of indexing methods to overcome the brute-force exhaustive matching with a more expensive distance is found in the CBID literature, e.g: in [21]. Haitsma *et al.* proposed an index of possible pieces of a fingerprint that points to the positions in the songs. Provided that a piece of a query's fingerprint is free of errors (exact match), a list of candidate songs and positions can be efficiently generated to exhaustively search through [4]. In [6], heuristics similar to those used in computational biology for the comparison of DNA are used to speed up a search in a system where the fingerprints are sequences of symbols. Kurth *et al.* [12] present an index that use code words extracted from binary sequences representing the audio. These approaches, although very fast, make assumptions on the errors permitted in the words used to build the index which could result in false dismissals. As demonstrated in [22], in order to guarantee no false dismissals, the simple (coarse) distance used for discarding unpromising hypothesis must lower bound the more expensive (fine) distance.

VI. Hypothesis Testing

This last step aims to answer whether the query is present or not in the repository of items to identify. During the comparison of the extracted fingerprint to the database of fingerprints, scores (resulting from distances) are obtained. In order to decide that there is a correct identification, the score needs to be beyond a certain threshold. It is not easy to choose a threshold since it depends on: the used fingerprint model, the discriminative information of the query, the similarity of the fingerprints in the database, and the database size. The bigger the database, the higher the probability of wrongly indicating a match by chance, that is a false positive. The false positive rate is also named false acceptance rate (FAR) or false alarm rate. The false negative rate appears also under the name of false rejected rate (FRR). The nomenclature is related to the Information Retrieval performance evaluation measures: Precision and Recall [1].

VII. Summary

We have presented a review of current trends of research in the recent area of audio fingerprinting. The different tasks involved in an audio fingerprinting system have been described. The purpose of each block has been commented along with some hints of the proposed solutions.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison Wesley, 1999.

[2] S. Subramanya, R.Simha, B. Narahari, and A. Youssef, "Transform-based indexing of audio data for multimedia databases," in *Proc. of Int. Conf. on Computational Intelligence and Multimedia Applications*, New Delhi, India, Sept. 1999.

[3] A. Kimura, K. Kashino, T. Kurozumi, and H. Murase, "Very quick audio searching: introducing global pruning to the time-series active search," in *Proc. of Int. Conf. on Computational Intelligence and Multimedia Applications*, Salt Lake City, Utah, May 2001.

[4] J. Haitsma, T. Kalker, and J. Oostveen, "Robust audio hashing for content identification," in *Proc. of the Content-Based Multimedia Indexing*, Firenze, Italy, Sept. 2001.

[5] M. Mihak and R. Venkatesan, "A perceptual audio hashing algorithm: a tool for robust audio identification and information hiding," in *4th Workshop on Information Hiding*, 2001.

[6] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust sound modeling for song detection in broadcast audio," in *Proc. AES 112th Int. Conv.*, Munich, Germany, May 2002.

[7] E. Allamanche, J. Herre, O. Helmuth, B. Fröba, T. Kasten, and M. Cremer, "Content-based identification of audio material using mpeg-7 low level description," in *Proc. of the Int. Symp. of Music Information Retrieval*, Indiana, USA, Oct. 2002.

[8] S. Sukittanon and L. Atlas, "Modulation frequency features for audio fingerprinting," in *Proc. of the ICASSP*, May 2002.

[9] S. Theodoris and K. Koutroumbas, *Pattern Recognition*. Academic Press, 1999.

[10] J. Picone, "Signal modeling techniques in speech recognition," in *Proc. of the ICASSP*, vol. 81, no. 9, Sept. 1993, pp. 1215–1247.

[11] J. Lourens, "Detection and logging advertisements using its sound," in *Proc. of the COMSIG*, Johannesburg, 1990.

[12] F. Kurth, A. Ribbrock, and M. Clausen, "Identification of highly distorted audio material for querying large scale databases," in *Proc. AES 112th Int. Conv.*, Munich, Germany, May 2002.

[13] G. Richly, L. Varga, F. Kovács, and G. Hosszú, "Short-term sound stream characterisation for reliable, real-time occurrence monitoring of given sound-prints," in *Proc. 10th Mediterranean Electrotechnical Conference, MEleCon*, 2000.

[14] C. Burges, J. Platt, and S. Jana, "Extracting noise-robust features from audio data," in *Proc. of the ICASSP*, Florida, USA, May 2002.

[15] T. Blum, D. Keislar, J. Wheaton, and E. Wold, "Method and article of manufacture for content-based analysis, storage, retrieval and segmentation of audio information," U.S. Patent 5,918,223, June, 1999.

[16] C. Papaodysseus, G. Roussopoulos, D. Fragoulis, T. Panagopoulos, and C. Alexiou, "A new approach to the automatic recognition of musical recordings," *J. Audio Eng. Soc.*, vol. 49, no. 1/2, pp. 23–35, 2001.

[17] E. Batlle, J. Masip, and E. Guaus, "Automatic song identification in noisy broadcast audio," in *Proc. of the SIP*, Aug. 2002.

[18] (2002) Etantrum. [Online]. Available: http://www.freshmeat.net/projects/songprint

[19] (2002) Musicbrainz trm. musicbrainz-1.1.0.tar.gz. [Online]. Available: ftp://ftp.musicbrainz.org/pub/musicbrainz/

[20] E. Chavez, G. Navarro, R. A. Baeza-Yates, and J. L. Marroquin, "Searching in metric spaces," *ACM Computing Surveys*, vol. 33, no. 3, pp. 273–321, 2001.

[21] S. Kenyon, "Signal recognition system and method," U.S. Patent 5,210,820, May, 1993.

[22] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *Proc. of the ACM SIGMOD*, Mineapolis, MN, 1994, pp. 419–429.