.

Content-based Classification and Retrieval of Audio Using the Nearest Feature Line Method

Stan Z. Li

School of EEE, Nanyang Technological University

BLK S1, Nanyang Avenue, Singapore 639798

szli@szli.eee.ntu.edu.sg http://markov.eee.ntu.edu.sg:8000/~szli/

Phone: +65 790-4540 Fax: +65 791-2687

Abstract

A method is presented for content-based audio classification and retrieval. It is based on a new pattern classification method called the nearest Feature Line (NFL). In the NFL, information provided by multiple prototypes per class is explored. This contrasts to the nearest neighbor (NN) classification in which the query is compared to each prototype individually. Regarding audio representation, perceptual and cepstral features and their combinations are considered. Extensive experiments are performed to compare various classification methods and feature sets. The results show that the NFL-based method produces consistently better results than the NN-based and other methods. A system resulting from this work has achieved the error rate of 9.78%, as compared to that of 18.34% of a compelling existing system, as tested on a common audio database.

Keywords

Audio classification, content-based retrieval, pattern recognition, nearest feature line (NFL).

EDICS Category: SA 3.2

I. Introduction

Audio data is an integral part of many modern computer and multimedia applications. Numerous audio recordings are dealt with in audio and multimedia applications. The effectiveness of their deployment is greatly dependent on the ability to classify and retrieve the audio files in terms of their sound properties or content. However, a raw audio signal data is a featureless collection of bytes with most rudimentary fields attached such as name, file format, sampling rate. This does not readily allow *content-based classification and retrieval.* While audio content may be described by using keywords and text, such information has so far been created manually. Rapid increase in the amount of audio data demands for a computerized method which allows efficient and automated content-based classification and retrieval of audio database [1], [2], [3], [4]. For these reasons, commercial companies developing audio retrieval products are emerging; see, for example, [1] (`http://www.musclefish.com`) and [5] (`http://www.comparisonics.com`).

Content-based classification and retrieval of audio sound is essentially a pattern recognition problem in which there are two basic issues: *feature selection*, and *classification* based on the selected features [6]. Concerning the former issue, an effective representation should be able to capture the most significant properties of audio sounds for the task, robust under various circumstances and general enough to describe various sound classes. For the latter issue, the formulation of a distance measure and the rule of classification are crucial.

While research in speech recognition, a closely related area, has a long history, research on content-based classification and retrieval of audio sounds is relatively new. Foster *et al.* [7] aim to allow queries such as “find the first occurrence of the note G-sharp”. Feiten and Ungvary [8] use a neural net to map sounds to text descriptions. Feiten and Günzel [9] use a self-organizing map (SOM) to group similar sounds based on perceptually-derived spectral features.

An important recent work is done by Wold *et al.* [1], represented by their system called “Muscle Fish”. The work distinguishes itself from the previous audio retrieval work [7], [8], [9] in its content-based capability. In the Muscle Fish system, various perceptual features, such as loudness, brightness, pitch, timbre are used to represent a sound. A normalized Euclidean (Mahalanobis) distance and the *nearest neighbor* (NN) rule are used to classify the query sound into one of the sound classes in the database. The technology, module and license of the Muscle Fish audio content-based retrieval have been adopted by Virage, Bulldog Group, BBC, Kodak, Mixman, Intraware, Opcode; see `http://www.musclefish.com/frameset.html` for more details.

In another work by Liu *et al.* [2], similar features plus sub-band energy ratios are used; the separability of different classes is evaluated in terms of the intra- and inter-class scatters to identify highly correlated features; and a classification is performed by using a neural network.

Foote [3] choose to use 12 mel-frequency cepstral coefficients (MFCCs) plus energy as the audio features. A tree-structured vector quantizer is used to partition the feature vector space into a discrete number of regions or "bins". Euclidean or Cosine distances between histograms of sounds are compared and the classification is done by using the NN rule. The best result is obtained with a supervised quantization tree with 500 bins, and a cosine distance measure.

In a work by Pfeiffer *et al.* [10], a filter bank consisting of 256 phase-compensated gammaphone filters proposed by Cook [11] is used to extract audio features. The audio signal is transformed into response probabilities. Such probability coefficients are used as audio features to classify audio content for applications such as audio segmentation, music analysis and violent sound detection.

In this paper, we present a new method for content-based audio classification and retrieval. The two aforementioned issues are addressed in the following way: Regarding feature selection, perceptual features, mel-cepstral features and their combinations are considered for the task (Section II). While perceptual features like brightness, bandwidth and sub-band energies capture the spectral characteristics of the sounds, some of characteristic features of sounds are lost. The cepstral coefficients capture the shape of the frequency spectrum of a sound, from which most of the original sound signal can be reconstructed, and hence provide a complement to the perceptual features.

Regarding the second issue, classification, a new pattern classification method, called the *Nearest Feature Line* (NFL) [12], [13], is used to explore information contained in the audio database (Section III). A basic assumption for the NFL is that a prototype (training) set of sounds are available and there exist more than one prototype (feature point) for each sound class, which is generally a valid assumption. The NFL makes use of information provided by multiple prototypes per class, in contrast to the commonly used NN in which classification is performed by comparing the query to each prototype individually.

The NFL works in the following way: Each pair of prototypes belonging to the same class are interpolated or extrapolated by a linear model. They are generalized by the *feature line* which is the line passing through the two points. The feature line provides information about variants of the two sounds, *i.e.* possible sounds derived from the two prototypes, and virtually provides an infinite number of prototype feature points of the class that the two prototypes belong to. The capacity of the prototype set is thus expanded. The classification is done by using the minimum distance between the feature point of the query and the feature lines.

Extensive experiments are performed to compare NFL, NN, and two others classification methods that make use of class information, and to compare various feature sets and their combinations (Section IV). The comparisons are based on two performance measures: classification error rate and retrieval score. The results show (i) that regardless of the feature set used, the NFL yields consistently better results than the other compared methods; (ii) that for the NFL, a combined feature set concatenating perceptual and cepstral features yields better performance than a perceptual or cepstral feature set alone. A system incorporating the NFL with a combined perceptual and cepstral feature set achieves the error rate of 9.78%, as opposed to that of 18.34% of the Muscle Fish system, as tested on a common database of 409 sounds. Demonstrations of the NFL and the NN classification and retrieval with the various feature sets can be accessed at `http://markov.eee.ntu.edu.sg:8000/~szli/demos.html`.

II. Audio Feature Selection

Before feature extraction, an audio signal (8-bit ISDN μ-law encoding) is preemphasized with parameter 0.96 and then divided into frames. Given the sampling frequency of 8000 Hz, the frames are of 256 samples (32ms) each, with 25% (64 samples or 8ms) overlap in each of the two adjacent frames. A frame is hamming-windowed by $w_i = 0.54 - 0.46 * \cos(2\pi i/256)$. It is marked as a silent frame if $\sum_{i=1}^{256}(w_i s_i)^2 < 400^2$ where s_i is the preemphasized signal magnitude at i and 400^2 is an empirical threshold. Then audio features are extracted from each non-silent frame. The means and standard deviations of the feature trajectories over all the non-silent frames are computed, and these statistics are considered as feature sets for the audio sound.

A. Definition of Audio Features

Two types of features are computed from each frame: (i) perceptual features, composed of total power, sub-band powers, brightness, bandwidth and pitch; and (ii) mel-frequency cepstral coefficients (MFCCs). Their definitions are given in the following, where the FFT coefficients $F(\omega)$'s are computed from the frame.

Total Spectrum Power. Its logarithm is used:

$$P = \log\left(\int_0^{\omega_0} |F(\omega)|^2 d\omega\right) \tag{1}$$

where $|F(\omega)|^2$ is the power at the frequency ω and $\omega_0 = 4000$Hz is the half sampling frequency.

Sub-Band Powers. The frequency spectrum is divided into 4 sub-bands with intervals $[0, \frac{\omega_0}{8}]$, $[\frac{\omega_0}{8}, \frac{\omega_0}{4}]$, $[\frac{\omega_0}{4}, \frac{\omega_0}{2}]$, and $[\frac{\omega_0}{2}, \omega_0]$. The logarithmic sub-band power is used

$$P_j = \log\left(\int_{L_j}^{H_j} |F(\omega)|^2 d\omega\right) \tag{2}$$

where L_j and H_j are lower and upper bound of sub-band j.

Brightness. The brightness is the frequency centroid

$$\omega_C = \frac{\int_0^{\omega_0} \omega|F(\omega)|^2 d\omega}{\int_0^{\omega_0} |F(\omega)|^2 d\omega} \tag{3}$$

Bandwidth. Bandwidth B is the square root of the power-weighted average of the squared difference between the spectral components and the frequency centroid

$$B = \sqrt{\frac{\int_0^{\omega_0} (\omega - \omega_C)^2 |F(\omega)|^2 d\omega}{\int_0^{\omega_0} |F(\omega)|^2 d\omega}} \tag{4}$$

Pitch Frequency. A simple pitch detection algorithm, based on detecting the peak of the normalized autocorrelation function, is used. The pitch frequency is returned if the peak value is above a threshold ($T = 0.65$, chosen empirically), or the frame is labeled as non-pitched otherwise.

Mel-Frequency Cepstral Coefficients. These are computed from the FFT power coefficients ([14], p.189). The power coefficients are filtered by a triangular bandpass filter bank. The filter bank consists of $K = 19$ triangular filters. They have a constant mel-frequency interval, and covers the frequency range of 0Hz – 4000Hz. Denoting the output of the filter bank by S_k ($k = 1, 2, \ldots, K$), the MFCCs are calculated as

$$c_n = \sqrt{\frac{2}{K}} \sum_{k=1}^{K} (\log S_k) \cos\left[n(k - 0.5)\pi/K\right] \qquad n = 1, 2, \ldots, L \tag{5}$$

where L is the order of the cepstrum. The MFCCs yield better results than the LPC cepstral coefficients.

B. Formation of Feature Sets

The means and standard deviations of the above 8 original perceptual features are computed over the non-silent frames, giving two feature vectors of 8-dimension each. The two vectors are concatenated to form a 16-dimensional vector. Adding the silence ratio (number of silent frames/total number of frames) and the pitched ratio (number of pitched frames/total number of frames) to this vector gives an augmented 18-dimensional perceptual feature vector, named "perc". Each x_i of the 18 components in the perc set is normalized according to $x_i' = (x_i - \mu_i)/\sigma_i$ (correlations between different features are ignored) where the mean μ_i and standard deviation σ_i are calculated over all the training set. This gives the final perceptual feature set, named "Perc".

Note the following differences between the perceptual features used in this work and in Muscle Fish: First, the two sets of perceptual features are different. Second, in Muscle Fish, there is no concatenation of the original features and their standard deviations into an augmented vector. Third, in Muscle Fish, the normalization is carried out in the calculation of the Mahalanobis distance by using the means and covariance matrix.

The means and standard deviations of the L MFCCs are also calculated over the non-silent frames, giving a $2L$-dimensional cepstral feature vector, named "Ceps$L$". In the experiments, Ceps$L$ with L values in the range between 5 and 120, with the corresponding feature sets named Ceps5, $\cdots$, Ceps120, are evaluated.

Note that the cepstral coefficients are not normalized. Empirically, the normalization of the perc set into Perc set by the mean and standard deviation gives better results whereas a similar normalization of the cepstral vectors leads to worse results.

The Perc and Ceps$L$ feature sets are weighted and then concatenated into still another feature set, named "PercCeps$L$", of dimension $18 + 2L$. The weighting is done as follows: There are 18 perceptual components in the Perc and $2L$ cepstral components in the Ceps$L$. Each of the 18 components has the unit standard deviation (std) after the normalization, and the total std of the 18 components is $s_1 = 18 \times 1$. The $2L$ components of the Ceps$L$ set are not normalized and have the total std of $s_2 = \sum_{i=1}^{2L} \sigma_i$ where σ_i is the std of the i-th component. To account for the relative reliability of the two sets, the two sets are weighted by $\frac{1}{s_1}$ and $\frac{1}{s_2}$, are concatenated into PercCeps$L$=$\frac{\text{Perc}}{s_1} \oplus \frac{\text{Ceps}L}{s_2}$ where $\oplus$ stands for the concatenation operation. This gives PercCeps5, $\cdots$, PercCeps120.

III. The Nearest Feature Line (NFL) Method

The rationale of the NFL is based on the following considerations: A sound corresponds to a point (vector) in the feature space. When one sound changes continuously to another in some way, it draws a trajectory linking the corresponding feature points in the features space. The trajectories due to changes between prototype sounds of the same class constitute a subspace representing that class. An audio sound of this class should be close to the subspace though may not necessarily be so to the original prototypes.

A. The Feature Line Space

Consider a variation in the sound space from point $\mathbf{z}_1$ to $\mathbf{z}_2$ and the incurred variation in the feature space from point $\mathbf{x}_1$ to $\mathbf{x}_2$. The degree of the change may be measured by $\delta\mathbf{z} = \|\mathbf{z}_2 - \mathbf{z}_1\|$ or $\delta\mathbf{x} = \|\mathbf{x}_2 - \mathbf{x}_1\|$. When $\delta\mathbf{z} \to \mathbf{0}$ and thus $\delta\mathbf{x} \to \mathbf{0}$, the locus of $\mathbf{x}$ due to the change can be approximated well enough by a straight line segment between $\mathbf{x}_1$ and $\mathbf{x}_2$. Thus any variant between the two can be interpolated by a point on the line. A further small change beyond $\mathbf{x}_1$ or $\mathbf{x}_2$ can be extrapolated using the linear model.

In the NFL method, a feature subspace is constructed for each class, consisting of the straight lines (feature lines) passing through each pair of the prototypes (feature points) belonging to that class. The straight line passing through $\mathbf{x}_1$ and $\mathbf{x}_2$ of the same class, denoted by $\overline{\mathbf{x}_1\mathbf{x}_2}$, is called a *feature line* (FL) of that class (see Fig.1). The FL provides information about linear variants of the two prototypes, *i.e.* possible sounds derived from the two.

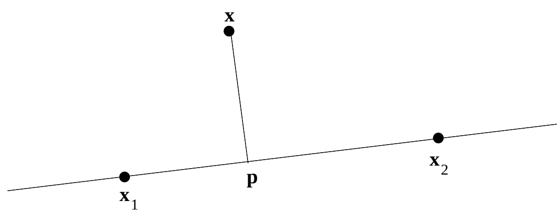

Fig. 1. Generalizing two feature points $\mathbf{x}_1$ and $\mathbf{x}_2$ by the feature line $\overline{\mathbf{x}_1\mathbf{x}_2}$. The feature point $\mathbf{x}$ of a query is projected onto the line as point $\mathbf{p}$.

Let $\mathbf{x}^c = \{\mathbf{x}_i^c \mid 1 \le i \le N_c\}$ be the set of the N_c prototypical feature points belonging to class c. A number of $K_c = \frac{N_c(N_c-1)}{2}$ lines can be constructed to represent the class. For example, $N_c = 5$ feature points are expanded by their $K_c = 10$ feature lines. The *FL space* for class c is composed of the K_c feature lines: $\mathbf{S}^c = \{\overline{\mathbf{x}_i^c\mathbf{x}_j^c} \mid 1 \le i,j \le N_c, i \ne j\}$, which is a subset of the entire feature space. When there are M classes in the database, M such FL spaces can be constructed, with a total number of N_{total} FL's where $N_{total} = \sum_{c=1}^{M} K_c$.

A feature line generalizes the original two feature points into an infinite many more points and so the FL space of a class is a much generalized representation than the individual prototypes, the generalization being done under the constraint of the original prototypes. The capacity of the prototypical set is thus expanded.

The feature vector as a function of variations in sound properties are highly nonconvex and complex. It constitutes a manifold, in general. The manifold can hardly be precisely described by a straight line in the feature space. To obtain a more accurate description of the variations, one may suggest that a higher order curve, such as splines, should be used [15]. This requires (i) that there should be at least three prototypical points for every class, and (ii) that these points should be ordered to account for relative variations described by only one parameter. For audio sound, requirement (ii) is difficult to meet; this is because the diversity among the prototypes is too complex and there are so many parameters that the prototypes cannot be sorted in a meaningful order for the construction of a spline manifold.

The NFL method generalizes the prototypes by constructing a simplified manifold, that is, the FL space. Although the FL space is a crude approximation for representing variations within an audio class, it turns out to be quite useful for the classification/retrieval purpose when used with the NFL criterion described below, and can achieve significant improvements over other conventional methods such as the NN.

B. Audio Classification and Retrieval Using NFL

The FL distance is defined below for audio classification and retrieval. Letting $\mathbf{p}$ be the projection point of the query $\mathbf{x}$ onto $\overline{\mathbf{x}_1\mathbf{x}_2}$ (see Fig.1), the *FL distance* from $\mathbf{x}$ to $\overline{\mathbf{x}_1\mathbf{x}_2}$ is defined as $d(\mathbf{x}, \overline{\mathbf{x}_1\mathbf{x}_2}) = \|\mathbf{x}-\mathbf{p}\|$ where $\|\cdot\|$ is some norm. The projection point can be computed as $\mathbf{p} = \mathbf{x}_1 + \mu(\mathbf{x}_2 - \mathbf{x}_1)$ where μ is a scalar, called the position parameter, can be calculated from $\mathbf{x}, \mathbf{x}_1$ and $\mathbf{x}_2$ as follows: Because $\overline{\mathbf{px}}$ is perpendicular to $\overline{\mathbf{x}_2\mathbf{x}_1}$, we have $(\mathbf{p}-\mathbf{x})\cdot(\mathbf{x}_2-\mathbf{x}_1) = [\mathbf{x}_1+\mu(\mathbf{x}_2-\mathbf{x}_1)-\mathbf{x}]\cdot(\mathbf{x}_2-\mathbf{x}_1) = 0$ where "$\cdot$" stands for dot product, and thus $\mu = \frac{(\mathbf{x}-\mathbf{x}_1)\cdot(\mathbf{x}_2-\mathbf{x}_1)}{(\mathbf{x}_2-\mathbf{x}_1)\cdot(\mathbf{x}_2-\mathbf{x}_1)}$. The parameter μ describes the position of $\mathbf{p}$ relative to $\mathbf{x}_1$ and $\mathbf{x}_2$. When $\mu = 0$, $\mathbf{p} = \mathbf{x}_1$. When $\mu = 1$, $\mathbf{p} = \mathbf{x}_2$. When $0 < \mu < 1$, $\mathbf{p}$ is an interpolating point between $\mathbf{x}_1$ and $\mathbf{x}_2$. When $\mu > 1$, $\mathbf{p}$ is a "forward" extrapolating point on the $\mathbf{x}_2$ side. When $\mu < 0$, $\mathbf{p}$ is a "backward" extrapolating point on the $\mathbf{x}_1$ side.

For NFL *classification*, a query feature point $\mathbf{x}$ is classified to class c if it is nearest to the FL space $\mathbf{S}^c$ of that class where the distance from $\mathbf{x}$ to $\mathbf{S}^c$ is the shortest distance from $\mathbf{x}$ to the FL's belonging to $\mathbf{S}^c$. For NFL *retrieval*, two patterns represented by $\mathbf{x}_i^c$ and $\mathbf{x}_j^c$ are retrieved as the top two if $\mathbf{x}$ is closest to $\overline{\mathbf{x}_i^c \mathbf{x}_j^c}$; other pairs can be retrieved and ranked according to the FL distance.

The NFL procedure consists of the follow steps: Calculate the FL distance between the query $\mathbf{x}$ and the feature line $\overline{\mathbf{x}_i^c \mathbf{x}_j^c}$ for each class c and each pair (i, j) where $i \neq j$. This yields a number of N_{total} distances. The distances are sorted in ascending order, each being associated with a class label c, two prototypes $\mathbf{x}_i^c$ and $\mathbf{x}_j^c$, and the corresponding μ value. The *NFL distance* is the first rank FL distance: $d(\mathbf{x}, \overline{\mathbf{x}_{i^*}^{c^*} \mathbf{x}_{j^*}^{c^*}}) = \min_{1 \leq c \leq M} \min_{1 \leq i < j \leq N_c} d(\mathbf{x}, \overline{\mathbf{x}_i^c \mathbf{x}_j^c})$. The first rank gives the information about the best matched class c^* for the NFL classification. The sorted list gives the retrieved sounds for the NFL retrieval.

IV. Experimental Results

The following experiments are aimed to evaluate (i) several classification and retrieval methods, that is, NFL, NN, k-NN and NC (nearest center), and (ii) three types of feature sets, namely Perc, Ceps, PercCeps. The results will also be compared with that of Muscle Fish [1] obtained from the search interface at `http://www.musclefish.com/cbrdemo.html`. An online demonstration of the present work can be accessed at `http://markov.eee.ntu.edu.sg:8000/~szli/demos.html`.

The k-NN is a decision rule for classification [6] while the NC can be used for both classification and retrieval. In NC, a class is represented by the center of the prototypes belonging to that class, and the distance between the query and a class is that between the query and the class center. The k-NN and NC are included in the comparison because they also make use of information about multiple prototypes per class as the NFL does but in different ways.

An audio database of 409 sounds from Muscle Fish is used for the experiments. The audio sounds are mono, 8 bit μ-law encoded and sampled at 8kHz, of a few seconds each, saved in the Next/Sun AU audio format. They are classified into 16 classes by Muscle Fish as shown in Table I.

Given a query sound and a set of prototype sounds, a classification/retrieval program returns a list of prototype sounds matched and sorted in the descending order of a distance. The following two measures will be used in performance evaluation:

1. *Error rate.* This is a performance measure for classification, defined as the ratio between the number of incorrect first rank matches and the total number of queries.

2. *Weighted score.* This is a performance measure for retrieval. First, define the weighted

TABLE I

409 SOUNDS OF 16 CLASSES IN THE AUDIO DATABASE

Sound Class c	# Sounds N_c	Sound Class c	# Sounds N_c
altotrombone	13	male	17
animals	9	oboe	32
bells	7	percussion	99
cellobowed	47	telephone	17
crowds	4	tubularbells	19
female	35	violinbowed	45
laughter	7	violinpizz	40
machines	11	water	7

score for a query q as

$$\eta(q,m) = \sum_{k=1}^{m} w_k \text{Match}(q, r_k) \tag{6}$$

where $r_1, \ldots, r_m$ are the m top ranked matches for the query q; $\text{Match}(q, r_k) = 1$ if r_k and q belong to the same class, or 0 otherwise; and $w_k = W \cdot \frac{1}{k}$ is a decreasing sequence of weights $(k = 1, 2, \ldots)$ where $W = 1/\sum_{k=1}^{N_q} \frac{1}{k}$ in which N_q is the number of available prototypes for the class that q belongs to. Because the weights w_k are decreasing with the rank position k, a higher ranked correct match contributes more to $\eta(q,m)$. The weights are normalized by the factor W in the following sense: When the top N_q matches are all correct, $\eta(q, N_q)$ reaches the highest possible value of 1. For a query set $\mathbf{Q}$, the average weighted score over all $q \in \mathbf{Q}$

$$\overline{\eta}(m) = \frac{1}{\#\mathbf{Q}} \sum_{q \in \mathbf{Q}} \eta(q,m) \tag{7}$$

is used as a performance measure where $\#\mathbf{Q}$ is the number of elements in $\mathbf{Q}$. This is a function of the number (m) of the considered top matches.

In the following, two sets of results are presented. In the first, each of the 409 sounds in the database is used as the query. In the second, the 409 sounds are partitioned into a prototype (training) set and a test set, and each of the sounds in the test set is used as the query.

A. Evaluation by Leave-One-Out Tests

In this set of experiments, each of the 409 sounds in the database is used as the query in turn. When a sound is used as the query, it is *not* used as a prototype, so the prototype set consists of the entire database minus the query. This is so called the "leave-one-out" test. If the query q belongs to class c, then there are $N_q = N_c - 1$ prototypes for that class and $N_{c'}$ for other classes

TABLE II

ERROR RATES (AND NUMBER OF ERRORS) OBTAINED BY USING LEAVE-ONE-OUT TEST

Feature Set	NFL	NN	5-NN	NC
Perc√	11.98% (49)	13.94% (57)	24.45% (100)	34.96% (143)
Ceps5	30.07% (123)	28.61% (117)	31.78% (130)	55.01% (225)
Ceps8	21.03% (86)	23.96% (98)	31.05% (127)	55.26% (226)
Ceps10	18.58% (76)	24.94% (102)	33.25% (136)	54.77% (224)
Ceps15	21.03% (86)	23.96% (98)	37.16% (152)	53.06% (217)
Ceps20	22.49% (92)	26.41% (108)	37.16% (152)	53.06% (217)
Ceps40√	16.87% (69)	22.98% (94)	28.36% (116)	42.05% (172)
Ceps60	18.09% (74)	23.96% (98)	30.07% (123)	41.56% (170)
Ceps80	16.38% (67)	22.98% (94)	28.61% (117)	42.05% (172)
Ceps100	16.87% (69)	24.45% (100)	29.10% (119)	42.54% (174)
Ceps120	16.87% (69)	23.47% (96)	28.36% (116)	42.30% (173)
PercCeps5	12.47% (51)	15.16% (62)	19.80% (81)	41.81% (171)
PercCeps8√	9.78% (40)	13.94% (57)	20.78% (85)	37.65% (154)
PercCeps10	11.74% (48)	17.60% (72)	22.49% (92)	36.92% (151)
PercCeps15	11.98% (49)	20.05% (82)	23.96% (98)	33.50% (137)
PercCeps20	13.45% (55)	21.03% (86)	24.94% (102)	32.03% (131)
PercCeps40	11.25% (46)	15.16% (62)	21.03% (86)	34.23% (140)
PercCeps60	12.47% (51)	14.91% (61)	22.74% (93)	33.99% (139)
PercCeps80	11.74% (48)	14.91% (61)	22.98% (94)	33.50% (137)
PercCeps100	12.96% (53)	14.91% (61)	22.00% (90)	33.99% (139)
PercCeps120	12.47% (51)	14.43% (59)	21.52% (88)	33.99% (139)

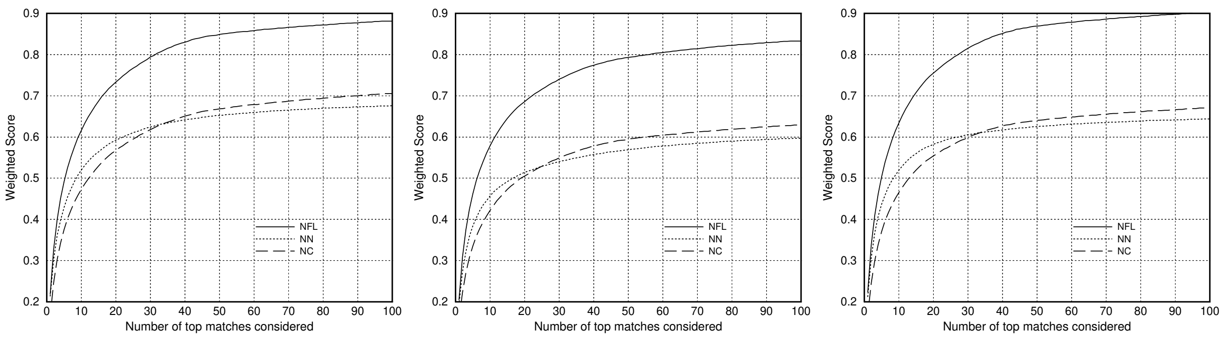

Fig. 2. Retrieval score functions $\overline{\eta}(m)$ of NFL, NN and NC for the Perc (left), Ceps40 (middle) and PercCeps8 (right) feature sets, obtained by using leave-one-out test on a single database.

$c' \neq c$. The output for a query is a list of best matches from the prototype set, sorted in the (NFL, NN, or NC) distance values.

Table II shows the error rates (and the numbers of errors in brackets) of the four classification methods and the three types of feature sets. The NFL yields consistently lower error rates than the other three methods for all the feature sets (except for Ceps5). Also, we see that the k-NN and NC methods are no better than the NN even though they use the class information.

Among all the Ceps feature sets, Ceps40 or Ceps80 is preferred over the others in terms of the error rate. For the NFL, the concatenation of the Perc and Ceps$L$ into PercCeps$L$ leads to improvements for most L values. The PercCeps8 feature set yields lower error rates than any perceptual or cepstral feature set alone. Overall, NFL+PercCeps8 yields the lowest error rate of 9.78%, of all combinations of methods and feature sets.

Fig.2 shows the retrieval performance of NFL, NN and NC measured in the weighted score as a function of the number of top retrieved sounds (k-NN has the same curve as NN), for the Perc, Ceps40 and PercCeps8 feature sets (ticked in the table). We can see from these curves that the NFL has consistently higher scores than the other methods.

B. Evaluation with Separate Training and Test Sets

In this set of experiments, the 409 sounds are partitioned into a prototype (training) set of 211 sounds and a test set of 198 sounds. During the test, each of the sounds in the test set is used as the query in turn. The output for a query is a list of best matched sounds from the prototype set, sorted in the (NFL, NN, or NC) distance values.

The partition is done in the following way: (i) Sort the sounds in each class in the alphabetical order of the file names, and then (ii) construct the two sets by including sounds 1, 3, $\cdots$ in the prototype set and sounds 2, 4, $\cdots$ in the test set. The 16 sound classes remain unchanged after the partition.

Recall that the "perc" feature vectors are normalized into "Perc" by the means and standard deviations. Here, the "perc" vectors of both training and test sets are normalized by the same means and standard deviations, that is, those computed from the "perc" of the *training* set.

Table III shows the error rates (and the numbers of errors in brackets) of the four classification methods and the three types of feature sets. Fig.2 shows the retrieval performance of NFL, NN and NC measured in the weighted score as a function of the number of top retrieved sounds (k-NN has the same curve as NN), for the Perc, Ceps40 and PercCeps8 feature sets (ticked in the table). These results are consistent with those obtained from the leave-one-out tests. Therefore similar conclusions can be drawn: (i) The NFL yields consistently lower error rates and higher weighted scores than the other methods for all the feature sets. (ii) For the NFL, the

TABLE III

Error rates (and number of errors) obtained by using separate training and test sets

Feature Set	NFL	NN	5-NN	NC
Perc$\surd$	14.65% (29)	16.67% (33)	23.23% (46)	35.35% (70)
Ceps5	28.28% (56)	29.29% (58)	33.84% (67)	53.54% (106)
Ceps8	21.72% (43)	26.77% (53)	30.81% (61)	54.04% (107)
Ceps10	18.69% (37)	21.21% (42)	30.30% (60)	55.05% (109)
Ceps15	19.19% (38)	23.23% (46)	30.30% (60)	53.54% (106)
Ceps20	20.71% (41)	24.24% (48)	32.32% (64)	52.53% (104)
Ceps40$\surd$	11.62% (23)	19.70% (39)	24.24% (48)	42.42% (84)
Ceps60	13.13% (26)	21.21% (42)	25.25% (50)	41.41% (82)
Ceps80	12.63% (25)	20.71% (41)	25.25% (50)	40.40% (80)
Ceps100	13.13% (26)	21.72% (43)	25.76% (51)	41.41% (82)
Ceps120	12.63% (25)	21.21% (42)	26.26% (52)	40.40% (80)
PercCeps5	12.12% (24)	17.68% (35)	21.21% (42)	43.43% (86)
PercCeps8$\surd$	9.60% (19)	13.13% (26)	22.22% (44)	38.89% (77)
PercCeps10	10.10% (20)	16.67% (33)	23.23% (46)	38.38% (76)
PercCeps15	13.64% (27)	17.17% (34)	22.22% (44)	34.34% (68)
PercCeps20	13.13% (26)	17.68% (35)	21.21% (42)	33.84% (67)
PercCeps40	11.62% (23)	15.15% (30)	21.72% (43)	32.83% (65)
PercCeps60	13.13% (26)	16.16% (32)	21.72% (43)	32.32% (64)
PercCeps80	12.12% (24)	15.66% (31)	20.71% (41)	32.83% (65)
PercCeps100	12.63% (25)	16.16% (32)	20.71% (41)	33.33% (66)
PercCeps120	12.63% (25)	17.17% (34)	20.71% (41)	33.33% (66)

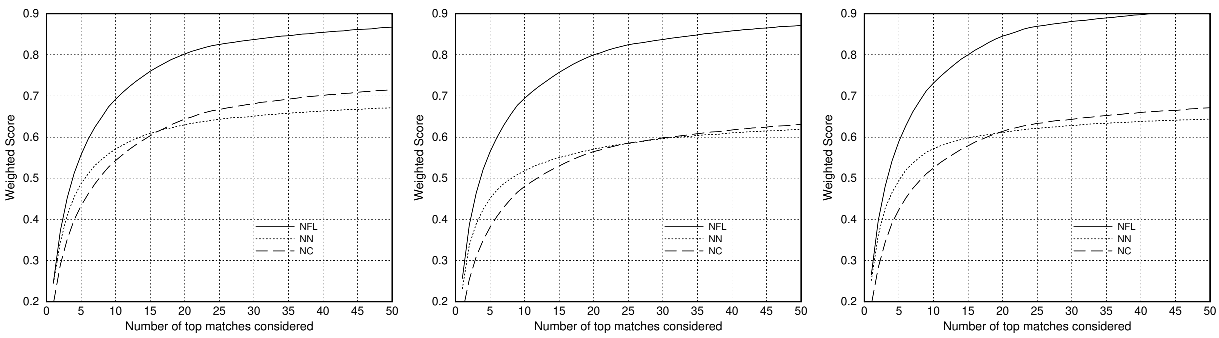

Fig. 3. Retrieval score functions $\overline{\eta}(m)$ of NFL, NN and NC for the Perc (left), Ceps40 (middle) and PercCeps8 (right) feature sets, obtained by using separate training and test sets.

concatenation of the two types of feature sets into PercCeps$L$ leads to improvements for most L values. (iii) The PercCeps8 feature set gives the best results of all the feature sets. (iv) The combination of NFL+PercCeps8 gives the overall best results of all methods and feature sets.

C. Comparison with Existing Systems

The Muscle Fish method [1] uses the NN rule and a perceptual feature set. Its classification error rate is 19.07% (78 errors out of 409 queries, see Table IV), as obtained from the Muscle Fish web interface `http://www.musclefish.com/cbrdemo.html` (the result files can also be obtained at `http://markov.eee.ntu.edu.sg:8000/~szli/Demos/Audio/MuscleFish`) which is roughly equivalent to the leave-one-out test with a modified version of NN. In comparison, the error rates of the leave-one-out test 13.94% (57 errors, see Table V) for the NN+Perc method and 9.78% (40 errors, see Table VI) for the NFL+PercCeps8 method, respectively. The lowest error rate of 9.78% (40 errors, see Table VI), obtained with NFL+PercCeps8, is significantly lower than that of Muscle Fish. Fig.4 compares the retrieval score curve of the three methods, which shows that NN+Perc has a performance similar to that of MuscleFish, and NFL+PercCeps8 is significantly better than MuscleFish, in terms of the score curve.

TABLE IV
78 CLASSIFICATION ERRORS MADE BY MUSCLE FISH

Sound Class	# Errors	Sound Class	# Errors
altotrombone	2	male	8
animals	4	oboe	5
bells	2	percussion	18
cellobowed	9	telephone	2
crowds	1	tubularbells	3
female	2	violinbowed	8
laughter	0	violinpizz	3
machines	5	water	6

TABLE V
57 CLASSIFICATION ERRORS MADE BY NN+PERC

Sound Class	# Errors	Sound Class	# Errors
altotrombone	1	male	7
animals	5	oboe	6
bells	1	percussion	12
cellobowed	3	telephone	1
crowds	0	tubularbells	1
female	4	violinbowed	3
laughter	0	violinpizz	1
machines	8	water	4

TABLE VI
40 Classification errors made by NFL+PercCeps8

Sound Class	# Errors	Sound Class	# Errors
altotrombone	1	male	7
animals	2	oboe	5
bells	0	percussion	2
cellobowed	3	telephone	1
crowds	0	tubularbells	0
female	4	violinbowed	3
laughter	0	violinpizz	1
machines	7	water	4

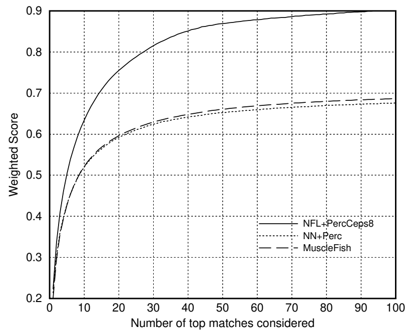

Fig. 4. Retrieval score functions $\overline{\eta}(m)$ of the three methods, calculated by using the leave-one-out tests.

In [3], a comparison is done between the Muscle Fish system and the Foote's system [3]. It is performed using 6 of the 16 Muscle Fish classes. The vector quantization is done by using all the sounds of the 6 classes including the test sounds themselves. The results show that those two systems have comparable performance, in terms of a measure called the "average precision" (AP). This indirectly suggests that the NN+Perc method should be better than Foote's method, and the NFL+PercCeps8 method significantly better, considering that the present work has been shown to yield significant better results than the Muscle Fish system in error rate.

V. Conclusion

The NFL makes use of available information of multiple prototypes within a class by constructing a subspace, for each class, that describe variations of features within a class. The experimental results show that given the same set of features, the NFL achieves consistently lower error rates and higher retrieval scores than the NN-type search methods. For the NFL, the concatenation of the perceptual and cepstral feature sets into PercCeps$L$ leads to improvements for most L values. Overall, the present method achieves the error rate of 9.78%, much lower than that of 18.34% of the Muscle Fish system, as tested on the the 409 sound database from the Muscle Fish.

The cost of the NFL spent on each class is proportional to the square of the number of prototypes for that class. A scheme has to be devised to reduce the cost when the number is large. One may propose to use a subset of the training data by sub-sampling. Questions are: what is a good strategy for the sampling and what is the incurred drop in the performance.

The experimental results here are obtained with small or moderate sizes of prototype (training) prototypes per class. A question is: how does the difference between NN and NFL change as the number of prototypes per class increases? It may be conjectured that NN performance should converge to that of NFL as the numbers approach to infinity. I would suggest that the topological shapes of the distributions are more crucial than these numbers; for example, an infinite number of co-linear prototypes gives the same NFL performance as two of them.

The NFL is a general pattern recognition method applicable when there are at least two prototypes per class. Recent research shows that the NFL yields better classification and retrieval performance than the NN also in other applications such as face recognition [12], [13], and image and texture classification and retrieval (unpublished). The NFL is empirically more powerful than the NN for the distributions in these applications. An investigation is being made to justify the NFL concept, and especially to find out classes of distributions for which the NFL performs better.

Acknowledgment – This work was supported by NTU AcRF projects RG 43/95 and RG 51/97. The author would like to thank the following people who helped this work: A reviewer suggested to use MFCCs which led to better results than the LPC cepstrum. Erling Wold provided information about implementation details of the Muscle Fish system. Jonathan Foote explained how to use the average precision (AP) scoring system. The use of the AP evaluation method were analyzed by Ong Chin Kiat. The implementation details and retrieval results of the Muscle Fish were obtained by Goh Ling Hwee, Mun Siong Yoong, Lee Kok Khuen, and Ng Chong Hai.

References

[1] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio", *IEEE Multimedia Magazine*, vol. 3, no. 3, pp. 27–36, 1996, http://musclefish.musclefish.com/ieeemm96/.

[2] Z. Liu, J. Huang, Y. Wang, and T. Chen, "Audio feature extraction and analysis for scene classification", in *IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, 1997, http://vision.poly.edu:8080/paper/audio-mmsp.html.

[3] J. Foote, "Content-based retrieval of music and audio", in *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, C. C. J. Kuo et al., Eds., 1997, vol. 3229, pp. 138–147.

[4] J. Foote, "An overview of audio information retrieval", *ACM-Springer Multimedia Systems*, 1998, In press.

[5] Stephen V. Rice, "Audio and video retrieval based on audio content", White pape, Comparisonics[tm], P.O. Box 1960, Grass Valley, CA 95945, USA, April 1998, http://www.comparisonics.com/WhitePaper.html.

[6] K. Fukunaga, *Introduction to statistical pattern recognition*, Academic Press, Boston, 2 edition, 1990.

[7] S. Foster, W. Schloss, and A. J. Rockmore, "Towards an intelligent editor of digital audio: Signal processing methods", *Computer Music Jounal*, vol. 6, no. 1, pp. 42–51, 1982.

[8] B. Feiten and T. Ungvary, "Organizing sounds with neural nets", in *Proceedings 1991 International Computer Music Conference*, San Francisco, 1991.

[9] B. Feiten and S. Günzel, "Automatic indexing of a sound database using self-organizing neural nets", *Computer Music Journal*, vol. 18, no. 3, pp. 53–65, 1994.

[10] S. Pfeiffer, S. Fischer, and elsberg W. E, "Automatic audio content analysis", Tech. Rep. No. 96-008, University of Mannheim, Mannheim, Germany, April 1996, ftp://pi4.informatik.uni-mannheim.de/pub/techreports/1996/TR-96-008.ps.gz.

[11] M. P. Cook, *Modelling Auditory Processing and Organisation*, Cambrige University Press, 1993.

[12] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method", *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 439–443, March 1999.

[13] S. Z. Li, "Face recognition based on nearest linear combinations", in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA., June 23-25 1998, pp. 839–844.

[14] Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of speech recognition*, Prentice Hall, Englewood Cliffs, N.J., 1993.

[15] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance", *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.