ARTHUR: Retrieving Orchestral Music by Long-Term Structure

Jonathan Foote
FX Palo Alto Laboratory, Inc.
3400 Hillview Avenue
Palo Alto, CA 94304
foote@pal.xerox.com

ABSTRACT

We introduce an audio retrieval-by-example system for orchestral music. Unlike many other approaches, this system is based on analysis of the audio waveform and does not rely on symbolic or MIDI representations. ARTHUR retrieves audio on the basis of long-term structure, specifically the variation of soft and louder passages. The long-term structure is determined from envelope of audio energy versus time in one or more frequency bands. Similarity between energy profiles is calculated using dynamic programming. Given an example audio document, other documents in a collection can be ranked by similarity of their energy profiles. Experiments are presented for a modest corpus that demonstrate excellent results in retrieving different performances of the same orchestral work, given an example performance or short excerpt as a query.

Keywords: music retrieval, audio analysis, acoustic similarity

1. INTRODUCTION

Recent years have seen an increasing interest in music retrieval by similarity. Because of the large amount of music available on the Web, there is starting to be significant commercial interest in music retrieval. Multiple start-up companies are offering audio-based music retrieval to internet users (for example, gigabeat.com and mongomusic.com). An intriguing business model is offered by "*CD" (starcd.com), which logs radio station playlists using automatic music identification. The company offers a service that lets users find the artist and title of a work (and, naturally, the opportunity to purchase it) based on the air time and the radio station ID.

The structure of most music is sufficient to characterize the work. As proof by example, human experts can identify music and sound by visual structure alone. Professor Victor Zue of MIT teaches a course in "reading" sound spectrographs. In a double-blind test, Arthur G. Lintgen of Philadelphia proved able to identify unlabeled classical phonograph recordings by the softer and louder passages visible in the LP grooves [1,2]. His example indicates that the long-term musical structure can be used for identification and retrieval. This paper presents a automatic music retrieval system inspired by Mr. Lintgen's approach, and is thus named ARTHUR in his honor. Like its namesake, ARTHUR retrieves audio on the basis of long-term structure, specifically the variation of soft and louder passages. The system thus works best on music that has substantial dynamic variation, such as works in the orchestral canon. Unfortunately, this technique is not robust for much popular music, which generally has much less dynamic range (a shortcoming shared with Mr. Lintgen).

Figure 1. Arthur G. Lintgen identifying a phonograph record by examining the grooves

2. PREVIOUS WORK

Much work in music retrieval has concentrated on symbolic or MIDI representations, perhaps due to the difficulty of extracting useful features from audio. Despite this, a growing number of researchers are investigating music and audio retrieval in the waveform domain [3]. A particular approach to rapid audio search was done by a group at NTT [4]. In this method known audio segments were detected in longer recordings by comparing histograms of the power spectrum in 7 frequency bands, and/or zero crossing rate. This method was optimized for speed, and could locate signals in the presence of noise, but relied on the identical signal being present in the search corpus.

Work at Muscle Fish LLC has resulted in a audio retrieval-by-similarity demonstration for small audio clips[1]. Muscle

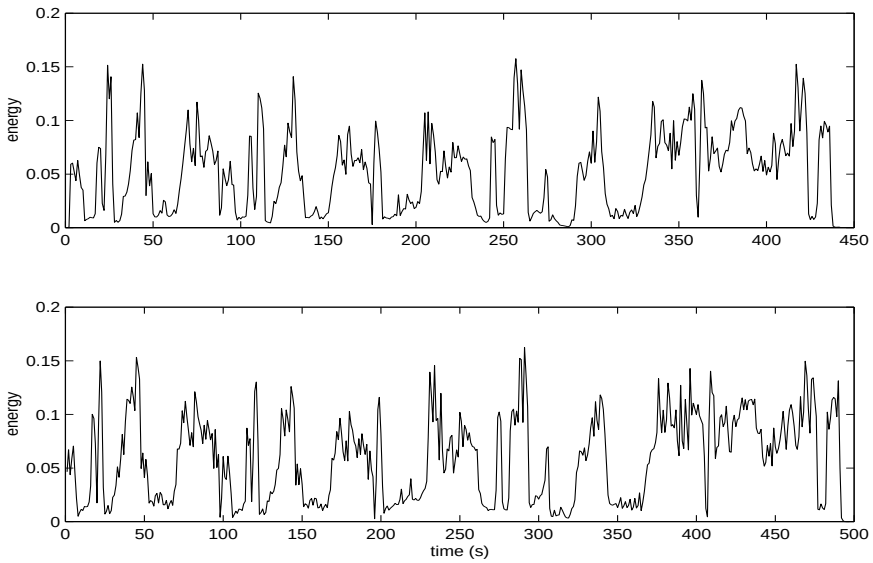

Figure 2. Energy profiles for two different performances of the first movement of Beethoven's *Fifth Symphony* Top: Herbert von Karajan/Berliner Philharmoniker. Bottom: Eric Leinsdorf/Boston Symphony Orchestra.

Fish's feature set includes loudness, pitch, bandwidth and harmonicity [5]. A Gaussian model is constructed from training data, so that a covariance-weighted Euclidean (Mahalonobis) distance can be used as a measure of similarity. For retrieval, the distance is computed between a given sound example and all other sound examples (about 400 in the demonstration). Sounds are ranked by distance, with the closer ones being more similar.

Work by the author, using an entirely different approach, has resulted in a similar retrieval application [6]. Here distance measures are computed between histograms derived from a discriminatively-trained vector quantizer. A histogram is computed for each audio file by counting the relative frequencies of samples in each quantization bin. If histograms are considered vectors, then simple Euclidean or cosine measures can determine the similarity, and thus rank the audio. Unlike previous approaches, this works for multicomponent audio sources such as music[1]. David Pye at ATT UK Research has developed another audio retrieval method using Gaussian models [7] that improves on the vector-quantizer approach in many respects.

3. THE ALGORITHM

The retrieval algorithm is relatively straightforward. First, an "energy profile" is computed for every audio document in the collection. The energy profile is a representation of the average acoustic energy versus time. In the experiments of the next section, this is determined by computing the RMS signal power across 1-second windows. Audio source files were obtained from CD recordings in 16-bit 44.1kHz stereo PCM format. To facilitate computation, files were mixed to mono and decimated to a 22.05 kHz sampling rate. An even more practical system could derive the audio power directly from encoded audio formats without the expense of decoding [7]. Though the similarity calculation is reasonably robust to scale changes, energy measurements are normalized so the maximum value is the same across all audio documents. The result of the analysis is a 1-d time series of power measurements at a rate of one per second. Figure 2 shows plots of energy versus time for two different performances of Beethoven's *Fifth Symphony.* Though the performances are clearly different (notice the different time scales), the overall energy structure of the documents is quite similar. This property is exploited by the ARTHUR system.

Once the energy profile is computed, it can be compared with other profiles. The similarity between them is calculated using dynamic programming (DP). Because DP is well-documented in the literature [8,9,10], the algorithmic details will only be summarized here. The particular variant used here is often called "Discrete Time Warping" in the speech recognition literature. This was originally developed for template-based speech recognition, where it helps account for variations in speech timing and pronunciation. One string is aligned to the other via a lattice, with the extent of one "test" signal on the vertical axis and the other "reference" signal on the horizontal. Every point (*i*,*j*) in the lattice corresponds to the alignment of the reference signal at time *i* to the test signal at time *j*. The DP algorithm first

[1]http://www.musclefish.com

[1]The reader is invited to try the demonstration at http://www.fxpal.xerox.com/people/foote/musicr/

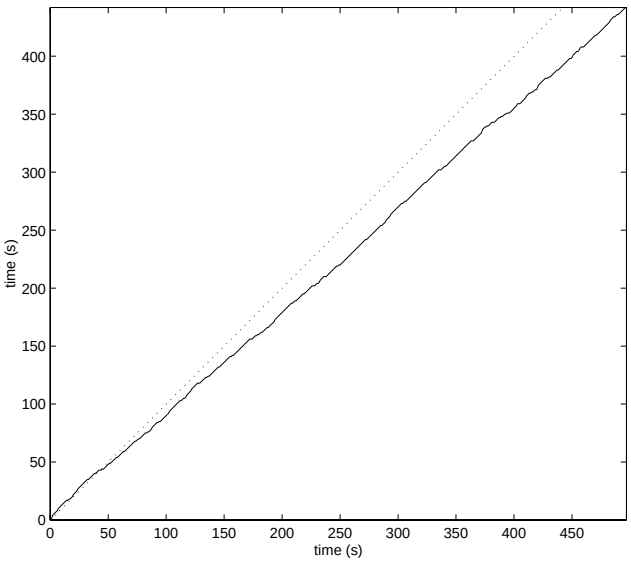

Figure 3. Dynamic-programming best-path alignment of energy profiles from Figure 2. Note deviation from diagonal (dotted line) due to performance differences.

recursively computes the best possible path through all points in the lattice. At every point, the cost of extending the path is calculated as the minimum of the cost of the best path (so far), plus the cost of extending the path. The latter cost is simply the distance between the test and reference signals at (*i*,*j*), plus a penalty for insertions or deletions. The latter are permitted by considering paths from neighbors not on the diagonal; in the current system paths from the nearest left or bottom neighbor are permissible. A penalty is added to the cost to discourage excessive insertions or deletions. Besides the cost of the best path, a pointer to the previous best path is also stored at each point. Once the best paths have been computed, the minimum-cost path is selected by minimizing over the last row of the lattice: this is the cost of the best-matching path. The actual path trajectory can be determined by backtracking using the pointers saved during the forward computation, as in Figure 3.

The DP algorithm returns two results: the best alignment path that takes one signal into the other, and the matching cost of that path. The last is an excellent measure of signal similarity: identical signals will have a diagonal best path and a cost of zero, while increasing differences will increase the matching cost. For retrieval, the cost is used to rank corpus documents by similarity to the query.

The DP algorithm is especially well suited to matching energy profiles. Unlike simple matching (as in [4]) or correlation, which require relevant documents to be exact replicas of the query, DP accounts for differences in both the features and the relative timing. In other words, signal amplitudes need not match exactly, nor is it required for the various features to occur exactly at the same relative times. Thus the DP algorithm smoothly matches performances with variable dynamics, tempos, and tempo changes. Figure 3 shows the best alignment path between the two Beethoven performances of Figure 2. Deviations from the diagonal show the relative tempo differences: overall the longer work has a slower tempo, except at the very beginning where it is slightly faster. In addition, DP easily handles the case when query and corpus files do not start and end at exactly the same time. Allowing insertions and deletions accounts for any offsets at the beginning or end. This feature is particularly useful for IR, because it allows queries to be any shorter fragment of longer works. Note that this application of dynamic programming is far from novel: as one of the earliest approaches to automatic speech recognition, it has been in use for more than three decades [10]. However, the features used here are rather different from a speech application in that they use a much longer time scale.

4. EXPERIMENTS

This paper presents results using an extremely modest corpus of less than 100 documents. Thus the experiments here are more to demonstrate the feasibility of the approach than to offer any convincing retrieval results. Many aspects of the system could likely be improved. In particular, there is a wide space of parameters still to be explored; for example the time window of 1 second for the energy calculation was chosen arbitrarily, and is likely to be suboptimal.

4.1 Experiment I: symphonic music

The corpus for the first experiment contained 58 documents, each of which was a single track from a CD recording. Three versions of Brahms' *Symphony No. 3*, including performances from two different conductors (Furtwangler and Celibidache), provide the queries and relevant documents for the experiment (see Table 1 for performance details). Other corpus documents were movements from Beethoven's Fifth, Sixth, and Seventh symphonies, including two separate performances of Beethoven's *Fifth Symphony.* A "Classics Greatest Hits" collection provided yet a third performance of the *Fifth's* first movement, as well as fifteen of the usual classical warhorses, including the *Allegro* movements from Bach's *Brandenburg Concerto No. 3* and *Eine Kleine Nachtmusik*, as well as an excerpt from the *William Tell Overture*, and similarly well-known works. The corpus was also "salted" with the nine tracks from Pink Floyd's *Dark Side of the Moon* (to speak of classic warhorses) plus two Beatles songs and four tracks from the John Coltrane album *Blue Train*. The queries were chosen from the Brahms symphony as each movement has two alternate performances that are considered relevant. The corpus thus contains highly relevant documents (different

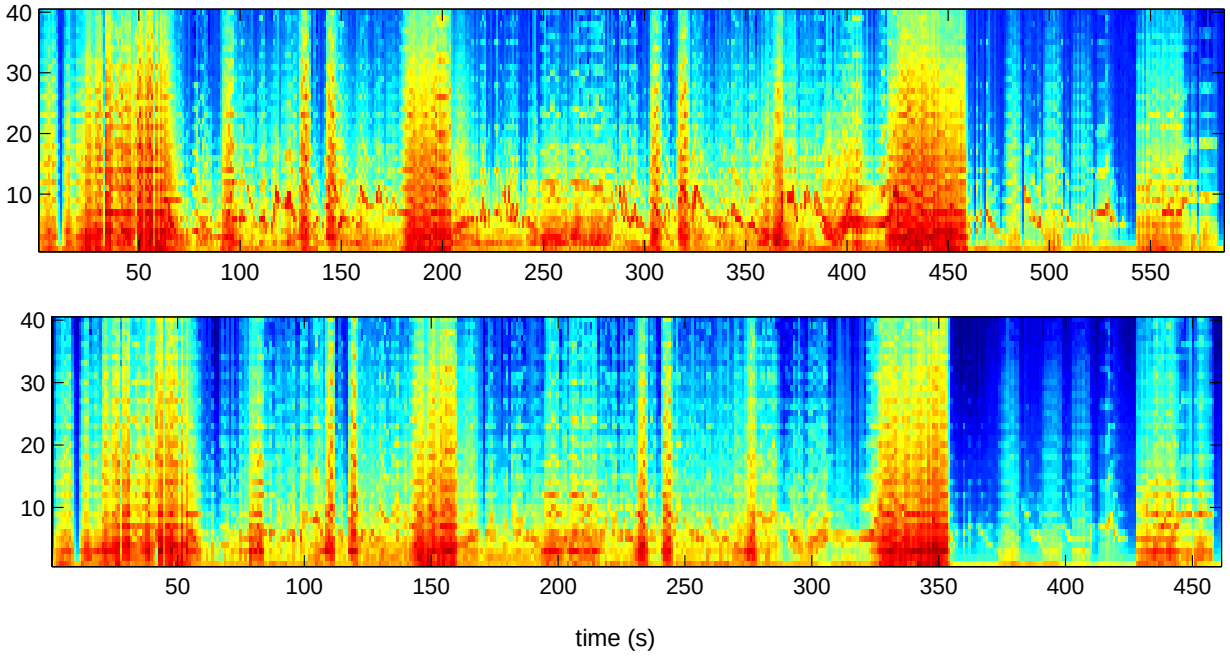

Figure 4. Spectrograms of different performances of the second movement of Beethoven's *Piano Concerto No. 2*. Top: Arthur Rubenstein/Erich Leinsdorf. Bottom: Levin/John Eliot Gardiner.

performances of the same movement), moderately similar documents (different movements from the same work), different works in a similar genre (Beethoven and Rossini), and non-relevant documents from unrelated rock and jazz genres (Pink Floyd and John Coltrane). As expected from the mostly orchestral genre, audio documents were rather long: the mean length of documents was 393 seconds with a range of 83 to 660 seconds. For experiments I, II, and III, entire audio documents, i.e. symphonic movements, were used as queries. For evaluation, different performances of the same movement were considered relevant, while different movements or works were not. The three performances of the first movement of Beethoven's *Fifth Symphony* were used to tune the retrieval algorithm, specifically the insertion/deletion penalties and the distance measure. The distance measure used was the squared Euclidean distance, and the insertion/deletion penalties were set to 0.1, which appeared to maximize the difference in DP scores between the relevant and non-relevant documents.

For the actual experimental evaluation, each of the three performances of the four movement of Brahms' *Third Symphony* (Op. 90 in F major) was used as a query. Each of the 58 corpus documents was then ranked by similarity to each of the 12 queries. For every query, the other two performances of the same movement ranked higher than any other document, thus yielding recall and precision rates of 100% on this corpus.

4.2 Experiment II: Piano concertos

Because the previous experiment proved suspiciously successful, it was desired to make the retrieval task more difficult, if for no other reason than to provide more credible results. Investigation revealed that piano music was not retrieved nearly as well as purely orchestral music. This is because the energy profile of piano music is highly variable between performances of the same work, even by the same performer. The acoustic energy is highly sensitive to both performance idiosyncrasies (such as use of the *sostenuto* pedal), the acoustic environment, microphone placement, recording post-production, and perhaps even the instrument make. For a more challenging retrieval task, the corpus of Experiment I was augmented with four performances of the three movements of the Beethoven *Piano Concerto No. 2* (Op. 19 in B flat major) as well as the Chopin *Concerto No. 2* (B 43/Op. 21 in F minor). The four performances of the six concerto movements resulted in a query set of 24 documents (see Tables 2 and 3 for performance information). These additional documents increased the overall corpus size to 82.

Date	Conductor	Ensemble
1954	Wilhelm Furtwängler	Berlin Philharmonic
1959	Sergiu Celibidache	Italian Radio Symphony
1979	Sergiu Celibidache	Munich Philharmonic

Table 1. Performances for Brahms query set

Date	Artist	conductor	Ensemble
1931	Artur Rubenstein	Sir John Barbirolli	London Symphony Orchestra
1946	Artur Rubenstein	William Steinberg	Pittsburgh Symphony Orchestra
1958	Artur Rubenstein	Alfred Wallenstein	Symphony of the Air
1968	Artur Rubenstein	Eugene Ormandy	Philadelphia Orchestra

Table 2. Piano query set: performances of Chopin's *Concerto No. 2* (B 43/Op. 21 in F minor)

Date	Artist	conductor	Ensemble
1956	Artur Rubenstein	Josef Krips	Boston Symphony Orchestra
1967	Arthur Rubenstein	Erich Leinsdorf	Symphony of the Air
1975	Artur Rubenstein	Daniel Barenboim	London Philharmonic Orchestra
1996	Robert Levin	John Eliot Gardiner	Orchestre Révolutionnaire et Romantique

Table 3. Piano query set: performances of Beethoven's *Piano Concerto No. 2* (Op. 19 in B flat major)

As expected, retrieval on this query set was gratifyingly poorer. Each of the 24 queries had 3 relevant documents in the corpus, so this was chosen as the cutoff point for measuring retrieval precision. Thus there were 72 relevant documents for this query set. For each query, documents were ranked by DP score, and a cutoff of 3 was used. From the 72 documents retrieved at this cutoff, 60 were relevant, giving a retrieval precision of 83%. More sophisticated analyses such as ROC curves, are probably not warranted due to the small corpus size. Retrieval performance for the original Brahms query set was not affected by the corpus expansion, and remained at 100%.

4.3 Experiment III: spectral features

Because the energy profile of piano music did not yield satisfactory performance, we attempted to improve retrieval by using features more informative than pure energy. Though many possible audio parameterizations are available, a spectral representation was chosen for its simplicity. For every audio document, a long-term spectral representation was computed using the Short-Time Fourier Transform. In the examples presented here, windows ("frames") are 1 second long. Each analysis frame is windowed with a Hamming window, and a fast Fourier transform (FFT) estimates the spectral components in the window. The logarithm of the magnitude of the result is used as an estimate of the power spectrum of the windowed frame. Because the comparatively long window has a high frequency resolution, the result was linearly quantized into 40 spectral bands ranging from 0 to $F_s/4$. For the 22.05 kHz data, this resulted in bands approximately 140 Hz wide. The resulting vector of 40 frequency components characterizes the spectral content of each 1-second window. The sequence of spectral vectors represents the frequency content of the signal over time (often called the spectrogram). Figure 4 shows spectrograms of two performances of the second movement of Beethoven's *Piano Concerto No. 2*. The spectrogram can be used in the dynamic programming in a similar manner to the energy. In this case the distance measure used is the squared Euclidean distance between spectral vectors. As in speech recognition, normalizing each document by subtracting the spectral mean improved retrieval considerably. Using spectral features resulted in 69 relevant documents retrieved out of the possible 72; thus the spectral features increased the retrieval performance from 83% to 96% on the piano query set. The retrieval performance on the original Brahms query set remained at 100% when using spectral features.

4.4 Experiment IV: variable query length

Using an entire audio track as a query seems to yield reasonable results, at least on this admittedly miniscule corpus. However, it might be desirable to use smaller audio clips as queries, if for no other reason than to speed up the search time (which is proportional to the product of the lengths of the query and corpus documents). Halving the query length reduces the overall search time by the same factor. To this end, the last experiment investigates retrieval accuracy as a function of query length. The queries were fragments of the piano queries from Experiment II formed by extracting a variable-length excerpt starting (arbitrarily) 40 seconds into each document. Once again, the DP algorithm can find the best match, regardless of where the clip starts or ends. Figure 5 shows the results of the experiment. As might be expected, longer queries perform better, and spectral features substantially outperform purely energetic features. Queries needed to be truncated at 130 seconds so as not to exceed the length of the shortest

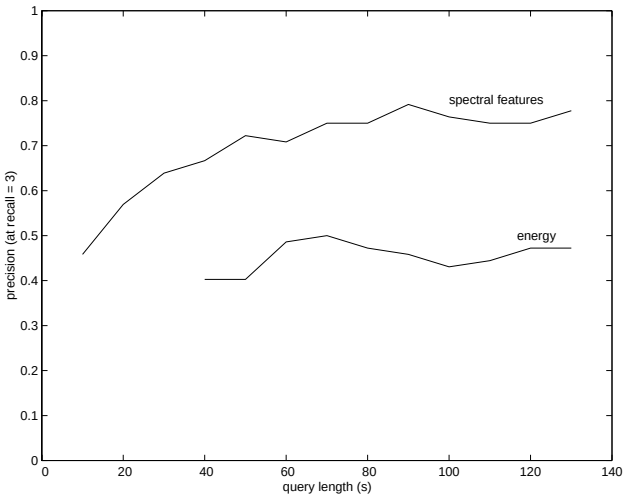

Figure 5. Retrieval performance versus query length, for piano query set (24 queries).

query document; this is one reason the best results in this experiment do not approach the precision achieved when using the full-length query documents. The non-monotonic results are no doubt due to the small test corpus: experiments on larger corpora should yield smoother curves.

5. DISCUSSION

These experiments are primarily a proof of concept given the admittedly small corpus size. There is considerable scope for improving the retrieval performance yet further. Tuning the algorithm on a larger development corpus should increase the differences between relevant and non-relevant document scores, and thus improve retrieval. Many aspects of the work here are arbitrary, such as the 1-second window size as well as the number of frequency bins, and also could be tuned. Better parameterizations might include a weighted frequency distance giving more importance to the middle frequencies, or even using cepstral features as in [7]. Obviously more evaluation on a bigger corpus would also not go amiss. However, we hope that these modest experiments have shown the utility of the approach.

6. ACKNOWLEDGEMENTS

Thanks to Stephen Smoliar for discussions and providing much of the test corpus data; also to Lynn Wilcox for manuscript suggestions. The photograph in Figure 1 was taken from reference [2], and is reproduced here under the Fair Use provision of the Copyright Act of 1976 (17 USCS § 107).

7. REFERENCES

[1] Holland, Bernard. “A Man Who Sees What Others Hear.” *The New York Times*. p. C28, 19 November 1981

[2] http://www.snopes.com/music/media/reader.htm

[3] Foote, J., “An Overview of Audio Information Retrieval,” in *Multimedia Systems*, **7**(1), pp. 2-11, January 1999, ACM Press/Springer-Verlag.

[4] Kashino, K., Smith, G., and Murase, H., “Time-Series Active Search for Quick Retrieval of Audio and Video,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 1999*, Phoenix, AZ. IEEE

[5] Wold, E., Blum, T., Keislar, D., and Wheaton, J., “Classification, Search and Retrieval of Audio,” in *Handbook of Multimedia Computing*, ed. B. Furht, pp. 207-225, CRC Press, 1999.

[6] Foote, J. “Content-Based Retrieval of Music and Audio,” in *Multimedia Storage and Archiving Systems II, Proc. SPIE*, Vol. 3229, Dallas, TX.

[7] Pye, D., “Content-based Methods for the Management of Digital Music,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2000*, vol. IV pp. 2437, IEEE

[8] J. Kruskal and D. Sankoff, “An Anthology of Algorithms and Concepts for Sequence Comparison,” in *Time Warps, String Edits, and Macromolecules: the Theory and Practice of String Comparison*, eds. D. Sankoff and J. Kruskal, CSLI Publications, (Stanford) 1999

[9] Rabiner, L., and Juang, B.-H., *Fundamentals of Speech Recognition*, Englewood Cliffs, NJ, 1993

[10] Vintsyuk, T. K., “Speech Discrimination by Dynamic Programming,” in *Kibernetika* **4**(2), pp. 81-88, Jan. 1968