QUICK AUDIO RETRIEVAL USING ACTIVE SEARCH

Gavin Smith *Hiroshi Murase* *Kunio Kashino*

NTT Basic Research Laboratories
3-1 Morinosato-Wakamiya, Atsugi-shi,
Kanagawa, 243-01, JAPAN
smith@apollo3.brl.ntt.co.jp, murase@apollo3.brl.ntt.co.jp, kunio@ca-sun1.brl.ntt.co.jp

ABSTRACT

This paper discusses a method to search quickly through broadcast audio data to detect and locate known sounds using reference templates, based on the active search algorithm and histogram modeling of zero-crossing features. Active search reduces the number of candidate matches between reference and test template by up to 36 times compared to exhaustive search, while still remaining optimal. Computation is further reduced by using computationally inexpensive zero-crossing features. The method is robust against white noise addition down to 20dB signal-to-noise ratios and digitization noise.

1. INTRODUCTION

This paper addresses the problem of detecting and locating sound objects from a stream of broadcast audio data quickly, while using computationally inexpensive processing. This has wide applications. One application is monitoring television audio data for the occurrence of a commercial. Another is the activation of a video cassette recorder (VCR) based on a programme's familiar theme tune alone, requiring no knowledge of the television timing schedule.

The emphasis of this paper is on increasing the speed of accurate audio retrieval using the active search algorithm, histogram modeling and computationally simple zero-crossing features. The active search algorithm has been applied to vision by Vinod and Murase [7]. In this paper we apply analogous methods to audio.

The paper is organized as follows. Section 2 presents the background theory. Section 3 explains the data used, experiments conducted and results. Section 4 is the discussion. Finally conclusions are presented in section 5.

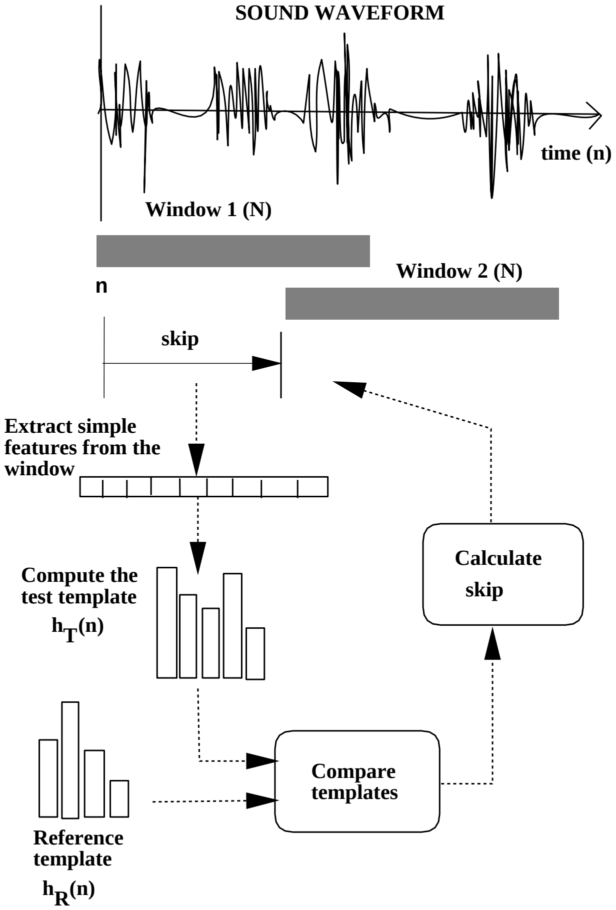
SOUND WAVEFORM
time (n)
Window 1 (N)
n
Window 2 (N)
skip
Extract simple features from the window
Compute the test template $h_T(n)$
Calculate skip
Compare templates
Reference template $h_R(n)$

Figure 1: Schematic diagram of the active search algorithm

2. THEORY

The general problem is that of searching for a reference template in a test audio stream. However the reference template will generally never be an exact match of any section of the test audio stream, because of noise. Noise may be due to different methods of digitization, thermal noise etc.

The method of search discussed in this paper is that of sliding the reference template across the test audio stream and evaluating the similarity between the reference template and the test audio stream at selected locations. Whenever the similarity rises above a threshold value, then the reference sound is detected and located. This requires a template model and a search algorithm. Both are investigated here. Figure 1 outlines the methods used.

2.1. Template modeling using histograms

The reference template is obtained by dividing the reference sound window into a number of fixed length frames, extracting a feature vector from each frame, and then finding the probability distribution of feature vectors in feature space over this window. A histogram is used as the non-parametric model for this distribution. The same process is applied to a window of test data to obtain a test template. Similarity between the test and reference templates h_R and h_T respectively, is achieved using histogram intersection, where B is the number of histogram bins:

$$S(h_R, h_T) \quad = \quad \sum_{i=1}^{B} min(h_R^i, h_M^i) \qquad (1)$$

Swain and Ballard [6] have shown that the histogram space provides sufficient inter-object discrimination in vision. This technique has been applied successfully to image object detection by Vinod and Murase [7]. We now apply these methods to the audio domain.

The choice of features is important. Cepstral coefficients have been used to distinguish speech from non-vocal sounds by Foote [1], and Hoyt and Wechsler [2]. However the cepstrum is computationally expensive and not suitable for quick searching.

Kedem shows that the zero-crossing rate (ZCR) and higher-order crossings of the time waveform discriminate sounds [4]. This is applied to real-time discrimination of speech from music by Saunders [5]. Further uses of the zero-crossing in speech are discussed by Juang and Rabiner [3]. The *i th* order zero-crossing Z_i over a sample N is defined as:

$$Z_i \quad = \quad \sum_{n=1}^{N} \frac{\left|sgn\big(s_i(n)\big) - sgn\big(s_i(n-1)\big)\right|}{2} \qquad (2)$$

$$sgn\big(s_i(n)\big) = \begin{cases} 1 & s_i(n) > 0 \\ -1 & s_i(n) < 0 \end{cases}$$

where s_i is the i th order difference signal:

$$s_i(n) = s_{i-1}(n) - s_{i-1}(n-1) \qquad (3)$$

These features are used because of their computational simplicity and proven discrimination properties.

2.2. The active search algorithm

The simplest method of search is that of sliding the reference template across the test audio stream one time step at a time and evaluating the similarity between the reference template and test audio stream at each time step. Whenever the similarity rises above a threshold value, then the reference sound is detected and located. This is termed *exhaustive* search and requires considerable computation at every time step.

However, similarity between the test and reference template shows considerable correlation from one time step to the next. The active search algorithm takes advantage of this by computing an upper bound on the similarity measure as a function of the time step, and skipping all intermediate time step similarity evaluations until this upper bound exceeds the detection threshold. Only then is the similarity measure evaluated, ignoring all intermediate evaluations, and thus reducing computation. The proof for the upper bound for the histogram model is given below, where frames are the time step units used.

Upper bound proof

h_R, $h_T(n_1)$ and $h_T(n_2)$ are the histograms for the reference template R and the test template T for frame numbers n_1 and n_2 respectively, normalized over the number of frames in each histogram, N. The histogram intersections at frames n_1 and n_2 are $S(h_R, h_T(n_1))$ and $S(h_R, h_T(n_2))$ respectively. Given the histogram intersection at frame n_1, the upper bound on the histogram intersection at frame n_2 can be calculated.

As the sliding test window moves forward in time from frame n_1 to n_2 one frame at a time, suppose every poorly matched frame between h_R and h_T leaves the window, and every new frame entering the window is a good match. This represents the maximum rate of increase of the histogram intersection. The upper bound on $S(h_R, h_T(n_2))$ is:

$${}_{ub}S\Big(h_R, h_T(n_2)\Big) \;=\; S\Big(h_R, h_T(n_1)\Big) + \frac{(n_2 - n_1)}{N} \quad (4)$$

$$(n_2 - n_1) = N\Big\{{}_{ub}S\Big(h_R, h_T(n_2)\Big) - S\Big(h_R, h_T(n_1)\Big)\Big\} \, (5)$$

If a correct match between reference and test template is defined when the histogram intersection exceeds some threshold S_{thresh} , then we need only evaluate the histogram intersection at time n_{crit}, and neglect all intermediate histogram intersections.

$$n_{crit} = N\left\{S_{thresh} - S\Big(h_R, h_T(n_1)\Big)\right\} + n_1 \quad (6)$$

This means that far fewer histogram intersections are evaluated compared to exhaustive search, where intersections are evaluated every time step. However the search algorithm still remains optimal insofar as detecting matches above a given threshold.

2.3. The audio retrieval algorithm

The audio retrieval algorithm is as follows:

1. Take the reference template histogram of N frames.
2. Obtain a test template histogram of the first N frames of the test audio stream, starting at frame 1. This is the seed model. Set $n = 1$.
3. Calculate the histogram intersection between the reference and test template $S(h_R, h_T(n))$ at frame number n. If this exceeds a threshold then the sound is detected.
4. Using equation 6 calculate the future frame number n_{crit} where the histogram intersection upper bound first exceeds the threshold.
5. Determine the test template shifted one frame along in time. This is achieved by subtracting the frame n feature vector from the histogram, and adding the frame $(N + n)$ feature vector to the histogram.
6. Set $n = n+1$.
7. If $n == n_{crit}$, then return to number 3. Otherwise return to number 5, until all the test data has been considered.

The test template is shifted one frame at a time in the direction of positive time. Each shift requires the evaluation of only two feature vectors. The histogram intersection between test template and reference template is only evaluated when the intersection upper bound exceeds the detection threshold. Multiple detections within one time window of the maximum intersection value are considered as a single sound object.

3. PROCEDURE AND RESULTS

Reference templates were selected uniformly from 121 seconds of 16 bit 44.1 kHz audio data, consisting of 10 commercials recorded from the television. Three different test audio data were considered.

1. **Test data 1** An exact copy of the reference audio data.
2. **Test data 2** The reference data was re-sampled using a different analogue to digital converter. This gave a 22dB SNR error of the reference relative to the test data, once both files were aligned and normalized to the same power. This models digitization noise, which is mainly a low frequency error.
3. **Test data 3** White noise was added to test data 2 at SNRs of 15dB, 20dB and 30dB relative to test data 2. This models digitization noise combined with transmission noise etc.

Both test and reference data were split into non-overlapping, non-windowed frames of 512 samples (11.6 ms). DC normalization was applied to each frame to reduce the effects of analogue-to-digital conversion DC offsets. In all experiments, both reference and test templates were pre-filtered with a 25-tap standard deviation 2 Gaussian-impulse response filter to reduce the effects of high frequency noise.

The histogram model was optimized with respect to the choice and size of feature vector, and the number, size and range of bins. Optimization involved a trade-off between performance and computational cost. The features used are the number of zero-crossings of the signal, first order and second order difference signal (Z_0, Z_1 and Z_2 respectively). Each dimension is divided into 8 equidistant bins ranging from 0 to 32, 0 to 48 and 0 to 88 for Z_0, Z_1 and Z_2 respectively.

Two sets of experiments were conducted. During each experiment, 120 reference templates were selected uniformly from the reference audio data. The results from the first 109 templates only were analyzed, because the remaining 11 templates sometimes occurred too close to the end of the test audio data for one entire window to be evaluated.

3.1. Testing model robustness and accuracy

The accuracy and robustness of the histogram model against noise for different window sizes was investigated using the exhaustive search algorithm. Experiments were conducted on test data 1,2 and 3. Table 1 shows the precision-recall results at the optimum threshold. The optimum threshold is defined when the sum of precision and recall is maximized.

3.2. Implementing the active search algorithm

The active search algorithm discussed in section 2.3 was tested. The optimum thresholds as determined in section 3.1 were used as the thresholds in the active

test data	window size	white noise SNR	recall	precision
1	11.89s		1.00	1.00
2	2.97s		0.94	0.96
	5.94s		0.99	0.97
	11.89s		1.00	1.00
3	2.97s	20dB	0.60	0.88
		30dB	0.90	0.91
	5.94s	20dB	0.78	0.87
		30dB	0.97	0.94
	11.89s	15dB	0.57	0.95
		20dB	0.87	0.99
		30dB	1.00	1.00

Table 1: Recall and precision ratios at the optimum thresholds

test data	window size	white noise SNR	ratio
2	2.97s		33.4
	5.94s		22.8
	11.89s		19.1
3	2.97s	20dB	35.8
		30dB	29.9
	5.94s	20dB	23.0
		30dB	20.1
	11.89s	15dB	20.5
		20dB	15.9
		30dB	13.0

Table 2: Ratio of the number of histogram intersection evaluations of exhaustive search relative to active search

search algorithm (see equation 6). The experiments in section 3.1 were repeated. Table 2 shows the ratio of the number of histogram intersection evaluations during exhaustive search relative to active search based on mean values.

4. DISCUSSION

Table 1 shows that the precision and recall ratios at the optimum threshold increases with increasing window length. This is because a larger window models a larger number of frames giving a histogram with a greater resolution. Hence different test templates can be well-discriminated. Moreover, the decision is being made using a greater amount of information. The 11.89s window shows the best precision-recall tradeoff. The tables show that the algorithm maintains useful recall-precision ratios down to 20dB SNR. The histogram model with zero-crossing features has acceptable robustness and accuracy for the applications in mind. The simple computation compared to cepstrum techniques increases search speed.

The use of the active search algorithm reduces the number of evaluations of the similarity measure between the test and reference templates relative to exhaustive search. Table 2 shows that the ratio can be as large as 36. Active search thus increases search speed.

5. CONCLUSIONS

The active search algorithm speeds up the search process, reducing the number of evaluations of the similarity measure between test and reference template up to a factor of 36 times, with no loss in optimality. A histogram model based on the zero-crossings discriminates sufficiently to be used as template models. Its simple computation also increases search speed. The template matching methods show robustness against white noise down to 20dB SNR and digitization noise.

Acknowledgments The authors wish to thank Dr Y Tohkura, Dr K Ishii and the members of the NTT Basic Research Labs for their help and encouragement in conducting this research.

6. REFERENCES

[1] JT Foote. A similarity measure for automatic audio classification. In *AAAI 1997 Spring Symposium on Intelligent Integration and Use of Text, Image, Video, and Audio Corpora*, March 1997.

[2] JD Hoyt H Wechsler. Detection of human speech in structured noise. In *ICASSP 94*, Vol. 2, pp. 237–240, 1994.

[3] B Juang L Rabiner. *Fundamentals of Speech Recognition.* Prentice Hall, Englewood Cliffs, New Jersey, USA, 1991.

[4] B Kedem. Spectral analysis and discrimination by zero-crossings. In *Proceedings of the IEEE*, Vol. 74, pp. 1477–1493, 1986.

[5] J Saunders. Real-time discrimination of broadcast speech/music. In *ICASSP 96*, Vol. 2, pp. 993–996, 1996.

[6] MJ Swain DH Ballard. Color indexing. In *International Journal of Computer Vision*, Vol. 7, pp. 11–32, November 1991.

[7] V V Vinod H Murase. Focused color intersection with efficient searching for object extraction. In *Pattern Recognition*, Vol. 30, 1997.