Systems and Computers in Japan, Vol. 34, No. 13, 2003
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J85-D-II, No. 10, October 2002, pp. 1552–1562

A Quick Search Method for Multimedia Signals Using Global Pruning

Akisato Kimura,[1] Kunio Kashino,[1] Takayuki Kurozumi,[1] and Hiroshi Murase[2]

[1]NTT Communication Science Laboratories, NTT Corporation, Atsugi, 243-0198 Japan

[2]Information Science Laboratories, Nagoya University, Nagoya, 464-8603 Japan

SUMMARY

The authors propose a new method for quickly searching for a specific audio or video signal to be detected within a long, stored audio or video stream to determine segments that contain signals that are nearly identical to the given signal. The Time-series Active Search (TAS) method is one of the quick search methods that have been proposed previously. This signal searching technique based on histograms extracted from the signals had implemented quick searching by local pruning, that is, omitting comparisons of segments for which searching was unnecessary based on similarities in the vicinity of the matching window. In contrast, the proposed technique implements significantly quicker searching by introducing global pruning, which looks at the entire signal time-series according to histogram classifications based on similarities of the entire signal to eliminate segments that need not be searched, in addition to local pruning. In this paper, the authors present a detailed discussion of the relationship between the degree of global pruning and the accuracy that is guaranteed. For example, the authors showed through experiments that when 128-dimension histograms were classified to 1024 clusters, the proposed technique achieved a search speed approximately 9 times that of TAS while preserving the same degree of accuracy. The preprocessing calculation time increased by approximately 1% of the time for playing the signal.

© 2003 Wiley Periodicals, Inc. Syst Comp Jpn, 34(13): 47–58, 2003; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/scj.10472

Key words: similarity search; Time-series Active Search method; global pruning; histogram intersection; L_2-distance.

1. Introduction

Lately, there has been a growing demand for techniques that enable a massive database of audio or video signals to be quickly and accurately searched for a specific audio or video signal.

Numerous techniques related to audio or video retrieval or searching have been proposed. Many of these techniques specify some kind of condition related to the contents of the audio or video signal to be detected to obtain specific audio or video signals that satisfy that condition from a database or long signal stream [1–5]. In this paper, this search method is referred to as content-based search.

On the other hand, in this paper, we propose a method for quickly and accurately searching for a specific audio or video signal to be detected (reference signal) within a massive stored audio or video stream (stored signal) to determine segments that contain signals that are nearly identical to the reference signal. In this paper, this search method is referred to as a similarity search. Similarity search techniques are widely applied to stored television or radio broadcasting data to detect or compile statistical information about specific commercials or tunes and to prevent the illegal use of music or video titles on the Internet.

The Time-series Active Search method, (hereafter TAS) which is a histogram-based signal search method, has been proposed as one of these techniques [6]. TAS implemented quick searching by omitting comparisons of segments for which searching was unnecessary based on similarities in the vicinity of the matching position. In this paper, this acceleration technique is referred to as local pruning. When feature extraction had been performed in advance, TAS enabled a segment identical to a 15-second audio or video fragment to be detected from 60 hours of stored audio or video signals within approximately 1 second. However, faster searching is required to search much more massive amounts of stored signals.

Since TAS uses only similarities in the vicinity of the matching point, even if a segment that is totally dissimilar to the reference signal continues for a long time within the stored signal, for example, matching cannot be completely omitted for that segment. As a result, a problem with TAS is that the search time increases according to the length of the stored signal, regardless of the similarity between the reference signal and entire stored signal.

Reducing the search range by taking into consideration similarities of the entire stored signal is indispensable for resolving this problem.

Many means of reducing the search range have been published mainly in fields related to database techniques for retrieving still images. For example, techniques that use hyperboxes [7], hyperspheres [8], or the common portions of hyperboxes and hyperspheres [9] for hierarchically classifying data in advance to construct a search tree for quickly retrieving similar images are typical. Since these techniques basically attempt to minimize the amount of search-time calculations regardless of the amount of preprocessing calculations, they are highly effective when the dimensions of the feature quantities extracted from the signal are relatively low such as for the retrieval of still images. However, these methods are not necessarily effective for audio or video searches, because the amount of calculations required for preprocessing increases explosively, or an enormous storage capacity is required to save the search tree as the dimensions of the feature quantities or the signal scale increases.

In contrast, in this paper, we propose a search technique that introduces global pruning for efficiently reducing the search range by taking into consideration similarities of the entire stored signal according to preprocessing that does not require a massive amount of calculations or a massive amount of storage in addition to the local pruning performed in the Time-series Active Search method. The remainder of this paper is organized as follows. Section 2 presents an overview of TAS. Section 3 describes the proposed method. Section 4 shows the effectiveness of the proposed method in experiments. Section 5 summarizes the paper.

2. Time-Series Active Search Method

2.1. Overview of the algorithm

Figure 1 shows an overview of the Time-series Active Search method. Since algorithm details can be found in Refs. 6 and 10, only important points are presented here.

First, feature vectors are calculated from both the reference signal and stored signal. Next, windows having the same length are applied to both the reference signal and stored signal to create histograms by classifying feature vectors within the windows. The existence of a reference signal is determined by whether or not a similarity value indicating the degree of similarity between the histograms exceeds a search threshold value that was set in advance. At this time, a time width (skip width) for which searching can be skipped in the time direction while guaranteeing that no false dismissals will occur can be obtained from the similarity and search threshold values, and searching proceeds by shifting the window that is applied to the stored signal forward in time by that width.

The feature vectors are calculated for both audio and video signals by using a similar method to the one described in Ref. 10. Histogram intersection [11] is used to measure the degree of similarity between histograms. Histogram intersection is defined by the equation

$$S_1 = S_1(H_R, H_S) \overset{\text{def}}{=} \frac{1}{D}\sum_{i=1}^{L} \min(h_{Ri}, h_{Si}) \quad (1)$$

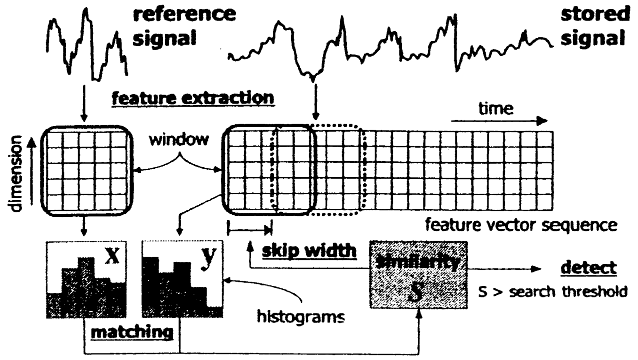


Fig. 1. Overview of the Time-series Active Search method.

where H_R and H_S are the histograms for the reference signal and stored signal, h_{R_i} and h_{S_i} are the numbers of feature vectors in the i-th dimension of each histogram, L is the number of histogram dimensions, and D is the number of feature vectors within the window.

The skip width w is obtained by using the following expression from a study related to the upper bound of the similarity values [6]:

$$w = \begin{cases} \text{floor}(D(\theta_1 - S_1)) + 1 & (\text{if } S_1 < \theta_1) \\ 1 & (\text{otherwise}) \end{cases} \quad (2)$$

where floor(x) means the greatest integer less than x, and θ_1 denotes the search threshold value.

Histogram intersection is used to measure the degree of similarity for the following reasons. (1) The calculation is easy. (2) The skip width is obtained by a simple calculation. (3) It provides a high search accuracy in a noisy environment.

2.2. Effectiveness of histogram intersection

To check the validity of histogram intersection as a measure of similarity, we used acoustic signals to perform experiments comparing histogram intersection and the L_2-distance (Euclidean distance) between histograms, which is generally used as a discrimination measure, with respect to the search accuracy. The L_2-distance is defined as follows:

$$d_2(H_R, H_S) \overset{\text{def}}{=} \sqrt{\sum_{i=1}^{L} |h_{Ri} - h_{Si}|^2} \quad (3)$$

First, we captured a 20-minute audio signal containing no repetitions in the computer two separate times. We let a fixed time segment extracted from a random location in one of these audio signals be the reference signal and let the other audio signal be the stored signal. Then, from each captured audio signal, we performed feature extraction according to the method described earlier. We also performed the experiment when white Gaussian noise was added to the stored signal.

In this experiment, we set the number of histogram dimensions and the SN ratio when noise was superimposed as parameters. We searched by repeating the Time-series Active Search method 100 times under the same experimental conditions and measured the search accuracy. We set the search threshold value for which the precision and the recall were equal and evaluated the accuracy by using the precision (= recall) at that search threshold value. The precision here is the percentage of correct matches among the matches that were output as search results, and the recall is the percentage of matches that were output as search results among matches that should be found.

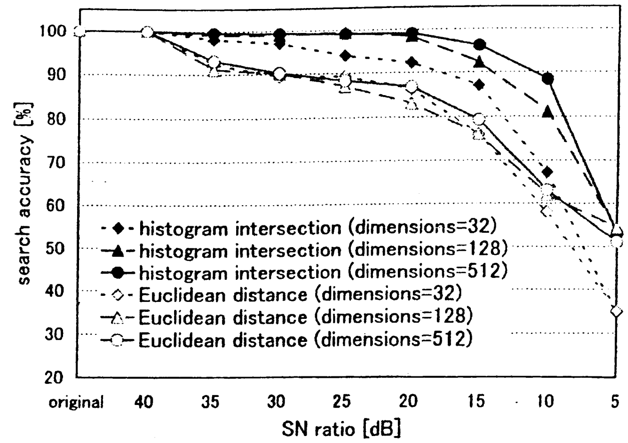


Fig. 2. Noise tolerances of two distance measures.

Figure 2 shows the experimental results. From this figure, it is apparent that at 35 dB or less, histogram intersection obtains a search accuracy that is approximately 10 to 15% higher than L_2-distance for every number of dimensions. If the number of dimensions is 128, histogram intersection obtains a search accuracy of at least 99% for SN ratios up to 20 dB.

From the above experiment, it is readily apparent that histogram intersection is superior to L_2-distance with respect to the search accuracy.

3. Proposed Method

3.1. Overview of the algorithm

Figure 3 shows an overview of the proposed method. In Fig. 3, the two rectangles represent the stored signal, and each division of the horizontal axis scale represents one signal frame.

With TAS, the maximum skip width is upperbounded by the number of the feature vectors within the window, or in other words by the window width, according to Eq. (2). However, with the proposed method, the entire stored signal is checked during preprocessing before the search is performed in order to find intervals within the stored signal that have a low degree of similarity to the reference signal. This enables the search range to be significantly reduced.

Figure 4 shows the processing procedure.

The processing is classified to preprocessing and searching. Preprocessing can be performed before a specific reference signal is assigned.

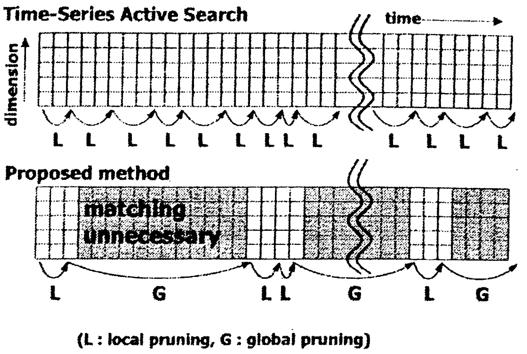

Time-Series Active Search
time
dimension
L L L L L L L L L L L L L
Proposed method
matching unnecessary
L G L L G L G
(L : local pruning, G : global pruning)

Fig. 3. Overview of the proposed method.

The preprocessing consists of the following four steps.

(1) Calculate feature vectors from the stored signal.

(2) Create histograms by applying the window to the feature vectors calculated in step (1) while sliding the window one frame at a time.

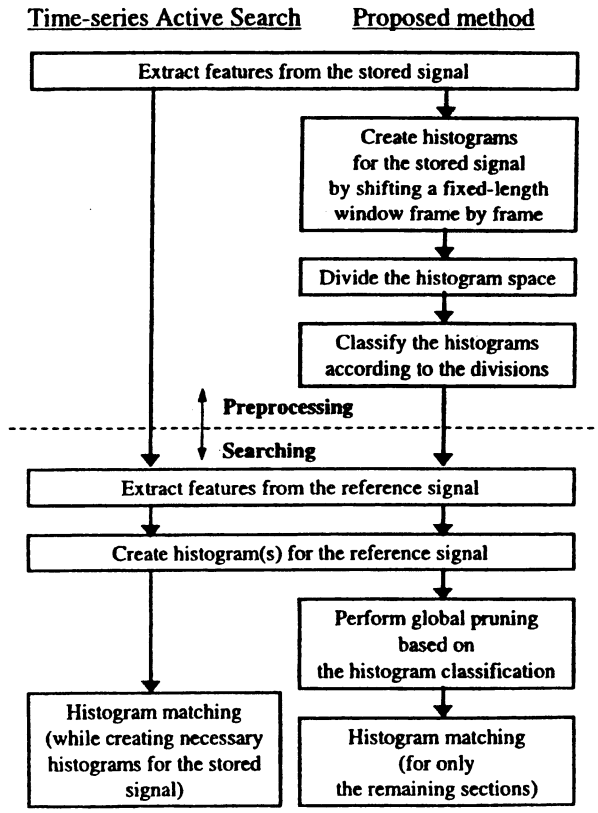

Time-series Active Search
Proposed method
Extract features from the stored signal
Create histograms for the stored signal by shifting a fixed-length window frame by frame
Divide the histogram space
Classify the histograms according to the divisions
Preprocessing
Searching
Extract features from the reference signal
Create histogram(s) for the reference signal
Perform global pruning based on the histogram classification
Histogram matching (while creating necessary histograms for the stored signal)
Histogram matching (for only the remaining sections)

Fig. 4. Processing procedure.

(3) Divide the histogram space.

(4) Classify the histograms according to the divided space.

Let the time window length be the assumed reference signal length (for example, 15 seconds). Also, the histograms are created by quantizing the feature vectors by using a certain vector quantization (VQ) algorithm and counting the number of feature vectors for each VQ codeword.

Hereafter, the histograms that were created from the stored signal will be referred to as the stored histograms, and the sequence of the stored histograms will be referred to as the stored histogram sequence. With TAS, the size of the window can vary for each given reference signal length since the stored histograms are not created in advance. However, with the proposed method, the size of the window is fixed because the stored histograms must be created in advance.

The search processing consists of the following four steps.

(1) Calculate feature vectors from the reference signal.

(2) Create histograms from the feature vectors calculated in step (1).

(3) Perform global pruning by using the histogram classifications.

(4) Perform histogram matching based on TAS only for histograms that belong to selected classes.

The histograms that were created from the reference signals will be referred to as reference histograms.

With TAS, stored histograms were created during histogram matching only at locations where matching was required. With the proposed method, since stored histograms are created at all locations during preprocessing, matching can also be performed during search processing by using these previously created stored histograms. However, this requires a large storage capacity to save the stored histograms in advance, and searching is difficult for a long stored signal. As a result, in this method, histograms are created during preprocessing only to obtain the time intervals that belong to each histogram classification. During search processing, new stored histograms are created at required locations during histogram matching in a similar manner as for TAS.

In addition to creating histograms for the saved signal prior to searching during preprocessing step (2), the proposed method classifies those histograms in advance during preprocessing steps (3) and (4). It can then perform search processing faster than TAS by reducing the search range during search processing step (3).

The following sections describe the main global pruning techniques in detail.

3.2. Dividing the histogram space

The LBG algorithm [12], for example, is used for the codebook training algorithm during vector quantization (VQ) when dividing the histogram space. The L_2-distance is used as the VQ distance measure.

3.3. Classifying histograms

Figure 5 shows an overview of the histogram classification scheme.

The VQ algorithm is used to classify each of the histograms that were created in advance for the stored signal. In other words, each histogram is assigned to the class having the representative histogram that is the minimum L_2-distance from that histogram. A representative histogram indicates one corresponding to a VQ code word. When the LBG algorithm is used for VQ codebook training, the representative histogram becomes the centroid of the training sample to which the same code word was assigned. The histogram classification created by the process described above is called a cluster.

Since the processing described in this section and the previous section does not require the reference signal, it can all be performed prior to searching.

3.4. Global pruning

3.4.1. Overview

The processing described in this section is performed after the reference signal is assigned. It uses the clusters that were obtained by the processing described in previous sections.

The reference histogram is classified according to the histogram classification scheme described in Section 3.3 into a cluster (a reference cluster) having a representative histogram at minimum L_2-distances. Next, clusters that satisfy a certain selection condition are selected, and histograms that are classified in the selected clusters are subject to searching. The stored histograms that must be searched are determined in this way, and histogram matching is performed based on TAS only for those histograms.

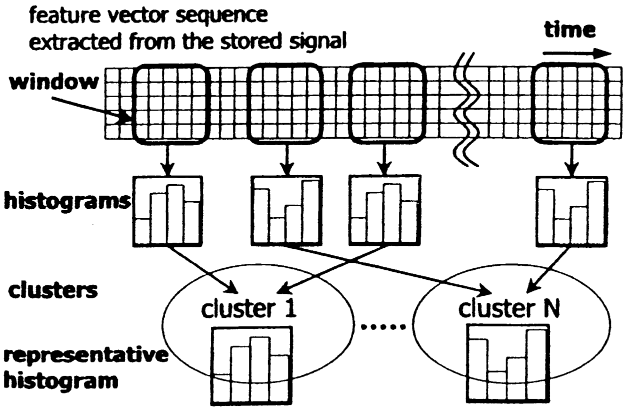


Fig. 5. Histogram classification scheme.

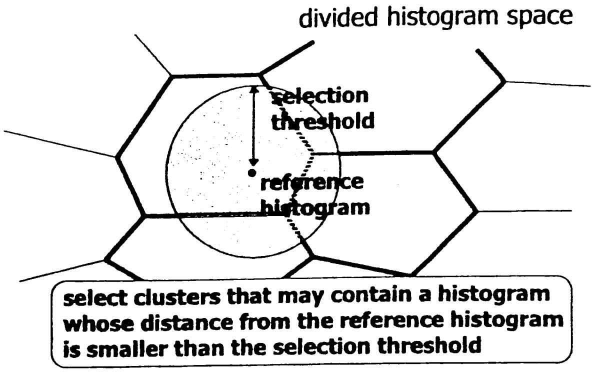

divided histogram space
selection
threshold
reference
histogram
select clusters that may contain a histogram
whose distance from the reference histogram
is smaller than the selection threshold

Fig. 6. Clusters selected according to global pruning.

The important point here is deciding how to specifically determine the cluster selection condition. This is described in the next section.

3.4.2. Cluster selection condition

As the cluster selection condition, we check whether or not a cluster can contain a histogram for which the L_2-distance from the reference histogram is less than the selection threshold, which is a predetermined value (Fig. 6). Only clusters that satisfy this condition are selected. Those that do not satisfy it are not selected. The decision equation for this selection condition is derived below.

First, we consider the clusters adjacent to the reference cluster.

Figure 7 represents a situation in which an L-dimensional histogram space is sliced by a plane on which the three points R, C_1, and C_2 reside, where R represents the reference histogram, C_1 the representative histogram of the reference cluster, and C_2 the representative histogram of a cluster adjacent to the reference cluster. d_{R1}, d_{R2}, and d_{12} indicate the L_2-distances between R and C_1, R and C_2, and C_1 and C_2, respectively.

The objective here is to derive an equation for determining whether or not there may exist a histogram for which the L_2-distance from the reference histogram exceeds the selection threshold value θ_2 among the histograms that belong to the cluster for which C_2 is the representative histogram.

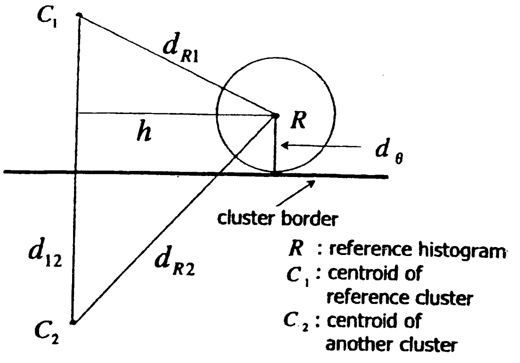

C_1
d_{R1}
h
R
d_θ
cluster border
d_{12}
d_{R2}
R : reference histogram
C_1 : centroid of reference cluster
C_2 : centroid of another cluster
C_2

Fig. 7. Cluster selection condition.

A histogram having a distance from the reference histogram R that does not exceed the selection threshold value θ_2, or in other words, a histogram that is inside a hypersphere of radius θ_2 centered on R, is subject to searching. Therefore, when the selection threshold value θ_2 is greater than the shortest distance d_θ between the reference histogram and the cluster, the cluster having C_2 as the representative histogram should be selected.

From Fig. 7, the following equations hold:

$$\begin{aligned} h^2 &= d_{R1}^2 - \left(\frac{1}{2}d_{12} - d_\theta\right)^2 \\ &= d_{R2}^2 - \left(\frac{1}{2}d_{12} + d_\theta\right)^2 \end{aligned} \tag{4}$$

The right-hand sides of these two equations can be equated and solved to obtain d_θ:

$$d_\theta = \frac{d_{R2}^2 - d_{R1}^2}{2d_{12}} \tag{5}$$

When a cluster is not adjacent to the reference cluster, the minimum distance between the cluster and the reference histogram is always greater than the value d_θ calculated by using Eq. (5). In other words, if clusters for which d_θ is less than or equal to the selection threshold value θ_2 are selected, every cluster that should be selected will be selected.

Therefore, according to the above discussion, clusters that satisfy the following equation should be selected:

$$\theta_2 \geqq \frac{d_{R2}^2 - d_{R1}^2}{2d_{12}} \tag{6}$$

3.4.3. Setting the threshold value based on upper and lower bounds of the similarity value

With the proposed method, two independent threshold values, namely, a search threshold and a selection threshold, must be set. The range in which accuracy is theoretically guaranteed varies according to the relationship between the search threshold value and selection threshold value. Therefore, we present guidelines below for deriving the relationship between the two threshold values and the range in which accuracy is theoretically guaranteed based on the relationship that holds between histogram intersection and the L_2-distance to determine the selection threshold value θ_2 from the search selection threshold value θ_1 so that the desired accuracy range is assigned.

In this context, "guaranteeing accuracy" means detecting all locations within the stored signal for which the similarity value (distance value) to the reference signal exceeds the threshold value (is less than the threshold value) and not detecting any location for which it is less than or equal to the threshold value (greater than or equal to the threshold value).

The L_1-distance

$$d_1 = d_1(H_R, H_S) \overset{\text{def}}{=} \sum_{i=1}^{L} |h_{Ri} - h_{Si}| \tag{7}$$

can be used to represent the histogram intersection [Eq. (1)] as follows:

$$S_1 = S_1(H_R, H_S) = 1 - \frac{1}{2D}d_1 \tag{8}$$

The relationship

$$\min(x, y) = \frac{1}{2}\{(x + y) - |x - y|\}$$

is used for the transformation from Eq. (1) to Eq. (8). Also, the following relationship generally holds between the L_1-distance and the L_2-distance:

$$d_2 = d_2(H_R, H_S) \leqq d_1 \leqq \sqrt{L}d_2 \tag{9}$$

From Eqs. (8) and (9), the following relationship is obtained:

$$\frac{2D}{\sqrt{L}}(1 - S_1) \leqq d_2 \leqq 2D(1 - S_1) \tag{10}$$

When the selection threshold value θ_2 satisfies

$$\theta_2 \geqq 2D(1 - \theta_1)$$

then from Eq. (10), all histograms that satisfy $S_1 \geq \theta_1$ will satisfy $d_2 \leq \theta_2$. In other words, all histograms that satisfy $S_1 \geq \theta_1$ can be selected without missing any even if global pruning is performed. On the other hand, when the selection threshold value θ_2 satisfies

$$\theta_2 \leq \frac{2D}{\sqrt{L}}(1 - \theta_1)$$

then from Eq. (10), all histograms that satisfy $d_2 \leq \theta_2$ will satisfy $S_1 \geq \theta_1$. In other words, histograms that satisfy $d_2 \leq \theta_2$ are always detected even if local pruning is performed.

As the size of the search threshold value θ_2 is reduced, the search range is significantly reduced due to global pruning, and the search speed can be expected to increase. However, reducing the search range too much creates a situation in which search misses easily occur. In addition, since the accuracy guarantee range approaches the accuracy guarantee range of the L_2-distance measure, the search accuracy is expected to decrease from the fact that the L_2-distance measure has a lower search accuracy than histogram intersection (Section 2.2).

Therefore, in this paper, we introduce the parameter p to define an equation for determining the selection threshold value from the search threshold value, and vary p to check the relationship between the search speed and search accuracy. The equation for determining the selection threshold value is

$$\theta_2 = \frac{2D}{(\sqrt{L})^p}(1 - \theta_1). \qquad (11)$$

Figure 8 shows how the range in which accuracy is guaranteed varies as p varies. As in Figs. 6 and 7, the histogram space is represented two dimensionally for simplicity.

The region in which the similarity to the reference histogram (histogram intersection value) exceeds the search threshold value θ_1 forms a regular 2^L-faced polyhedron in the L-dimensional space. For a two-dimensional space, this is a square as shown in Fig. 8. When $p = 0$, then $\theta_2 = 2D(1 - \theta_1)$, and the region where the (L_2) distance to the reference histogram is less than the selection threshold value θ_2 forms a hypersphere that exactly circumscribes the region in which the similarity to the reference histogram exceeds the search threshold value. In other words, the selection threshold value at $p = 0$ is the minimum selection threshold value for which accuracy is guaranteed based on the L_1-distance measure. Also, when $p = 1$, then $\theta_2 = (2D/\sqrt{L})(1 - \theta_1)$, and the region where the distance to the reference histogram is less than the selection threshold value θ_2 forms a hypersphere that is exactly inscribed in the region in which the similarity to the reference histogram exceeds the search threshold value.

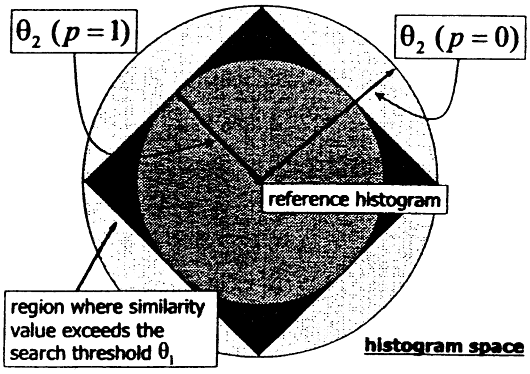


Fig. 8. Relationship between the search threshold value θ_1 and the selection threshold value θ_2 when $p = 0$ and $p = 1$.

According to the above investigation, p controls the radius of the hypersphere formed by the region in which the distance to the reference histogram is less than the selection threshold value. Therefore, we will refer to p as the radius parameter hereafter.

4. Experiments

4.1. Threshold value settings

To show the effectiveness of the proposed method, we measured both the time required to search for a specific 15-second audio signal from 200 hours of audio signal data and the search accuracy. The computer used for the experiments was a PC with a Pentium III 1-GHz CPU.

First, we captured the audio signal from a TV broadcast, encoded it in MP3 format, and recorded it on an external storage unit. We used MP3 format here because the storage size is smaller than for the source signal. In addition to a one-time capture of 200 hours for use as the stored signal, we similarly captured the audio signal from another TV station and recorded 1000 different 15-second signals for use as the reference signal. In both cases, we captured the source signal in monaural using a 32-kHz sampling frequency and linear 16-bit quantization accuracy, and we performed the MP3 encoding using a bit rate of 56 kbit/s.

We performed feature extraction from the captured audio signals by using the same method as used in Ref. 6. The feature vector time width was 60 ms, and the time step was 10 ms. We created histograms by using a codebook of size L, which was created in advance, for the vector quantization of each feature vector. We classified the sequence of stored histograms into C clusters according to the processing described in Sections 3.2 and 3.3, and we set the search threshold value θ_1 to 0.85.

To evaluate changes in the search accuracy and search speed due to changes in the radius parameter p, we varied p from 0 to 2 and measured the time required for searching and the search accuracy. In the experiments in this section, we assumed that the number of histogram bins was $L = 128$ and the number of clusters was $C = 1024$.

In the following discussion, all times are measured in terms of CPU time, and the average value when measurements were performed for 1000 reference signals is shown.

(1) Feature extraction time

The CPU time required to perform feature extraction from the 200-hour stored signal and 15-second reference signal while performing MP3 decoding was 7 hours, 34 minutes, and 35 seconds (approximately 4% of the playing time).

(2) Vector quantization time

The CPU time required to perform vector quantization of the feature vector sequences that were extracted from the 200-hour stored signal and 15-second reference signal was 12 minutes and 30 seconds (approximately 0.1% of the playing time). This was obtained by measuring the time for processing in memory after all feature vectors were loaded into memory.

(3) Histogram classification time

The CPU time required for classifying the histograms that were created from the 200-hour stored signal and 15-second reference signal was 1 hour, 30 minutes, and 37 seconds (approximately 0.8% of the playing time). This was obtained by measuring the time for processing in memory while sequentially creating the histograms by loading the sequence of feature vectors after vector quantization had been performed for all vectors.

(4) Pruning time

Global pruning is the search for the centroid of the cluster with the minimum distance from the reference histogram, the calculation of the selection condition decision equation (6), and the selection of classified histograms. The amount of calculations in the above procedure depends only on the number of histogram dimensions L and the number of clusters C. As a result, the pruning time shows a fixed value regardless of the size of the radius parameter p.

In the experiments in this section ($L = 128$ and $C = 1024$), the pruning time was 0.02 s. As shown later, this will be approximately 5 to 50% of the search execution time. However, if we ignore the operation for selecting preclassified histograms, the pruning time is also fixed regardless of the length of the stored signal. Therefore, it can be ignored when the stored signal is even longer.

(5) Search execution time

Figure 9 presents the experimental results.

Figure 9 shows results that compare the search execution times of TAS and the proposed method. The horizontal axis is the radius parameter p, and the vertical axes show the search execution time on the left and the search time reduction rate on the right. The search time reduction rate indicates the degree to which the proposed method was able to reduce the search execution time relative to TAS. It is defined as the ratio of the search execution time for TAS relative to the search execution time for the proposed method.

As shown in Fig. 9, when p is greater than 0.4, the search execution time for the proposed method gets shorter as the radius parameter increases. When $p = 1.0$, the search is executed in approximately 0.3 s, and when $p = 2.0$, it is executed in approximately 0.025 s. At this time, the proposed method can execute the search approximately 9 times faster ($p = 1.0$) or approximately 110 times faster ($p = 2.0$) than TAS.

From $p = 0$ to $p = 0.4$, the search execution time of the proposed method does not decrease monotonically as p increases. The reason for this is as follows.

As the radius parameter p increases, the amount of the stored signal that must be searched decreases monotonically. However, when a relatively large proportion of the entire stored signal is occupied by the stored signal that must be searched, there can often exist a large number of intervals that must be searched with short intervals between them. In this situation, when matching and skipping is performed based on TAS for each interval, the probability that a location that had been skipped by the conventional TAS method falls in the gap between the intervals that must be searched increases. In other words, the skip width may end up being reduced.

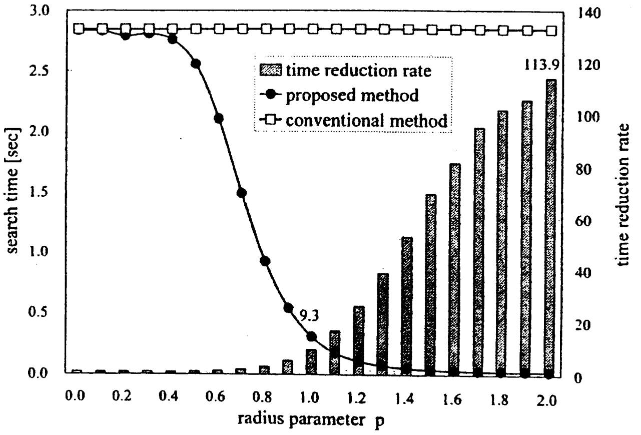


Fig. 9. Relationship between the radius parameter and the search time required for searching a stored audio signal of 200 hours.

Besides the time required for searching, time is also required for:

(6) Creating the codebook to be used for vector quantization of the feature vectors (codebook creation time), and

(7) Dividing the histogram space (histogram space division time).

However, if codebook creation and histogram space division are performed in advance by sampling only a sufficient number from among the various types of signals, these kinds of processing need not be executed again for each stored signal. As a result, in this paper, the codebook creation time and histogram space division time are not included in the time required for searching.

(8) Search accuracy

Figure 10 presents the experimental results.

Figure 10 shows the results of measuring the search accuracy for the proposed method. The horizontal axis is the radius parameter p, and the vertical axis the search accuracy. The search accuracy was evaluated according to the precision when the search results for TAS were assumed to be correct answers.

As shown in Fig. 10, the search accuracy begins to drop at p = 1.0, and the accuracy decreases to 85% at p = 2.0.

4.2. Search parameters

To check the relationship between the number of histogram dimensions L and number of clusters C and the search accuracy and search speed, we varied L and C and measured the time required for searching and the search accuracy. In this section, the audio signals that were used and the feature extraction parameters are the same as those used for the experiments in the previous section. For the search threshold value θ_1, we set $\theta_1 = 0.85$ (L = 128), θ_1 = 0.8 (L = 256), and θ_1 = 0.75 (L = 512). For the radius parameter p, we set p = 1.

Since (1) feature extraction time, (2) vector quantization time, and (3) histogram classification time are the same as described in the previous section, they are omitted here.

(4) Pruning time

Figure 11 shows the experimental results.

Figure 11 shows the results when the pruning time for the proposed method was measured with the number of histogram dimensions as a parameter. The horizontal axis is the number of clusters, and the vertical axis the pruning time.

As shown in Fig. 11, the pruning time for each number of dimensions increases almost linearly relative to the number of clusters. Also, the pruning time tends to increase as the number of dimensions increases. For example, when the number of dimensions is 128 and the number of clusters is 1024, the pruning time is 0.02 s.

(5) Search execution time

Figure 12 shows the experimental results.

Fig. 12 shows comparison results of the search execution times of TAS and the proposed method with the number of histogram dimensions as a parameter. The horizontal axis is the number of clusters, and the vertical axis the search execution time.

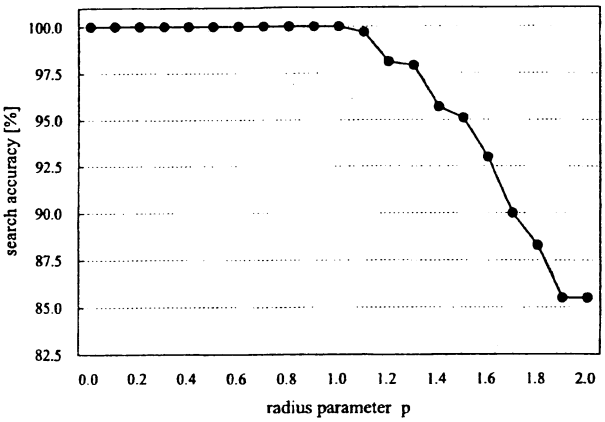


Fig. 10. Relationship between the radius parameter and search accuracy.

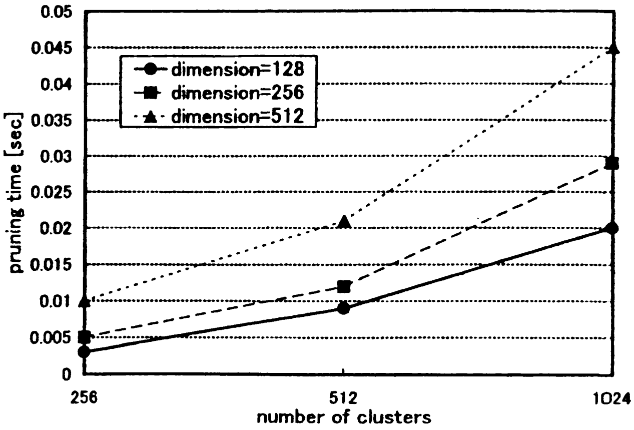


Fig. 11. Relationship between the number of clusters and pruning time for a stored audio signal of 200 hours.

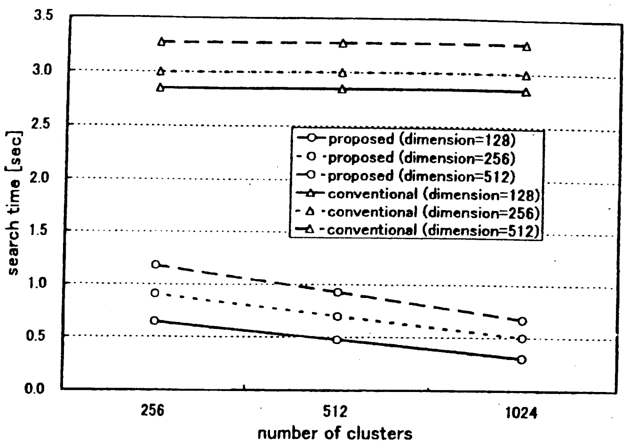


Fig. 12. Relationship between the number of clusters and search time for a stored audio signal of 200 hours.

As shown in Fig. 12, the search execution time for the proposed method decreases as the number of clusters increases, and when the number of clusters is 1024, the search is executed in less than 1 second for every number of dimensions. Also, with both methods, the search execution time tends to increase as the number of dimensions increases.

(6) Search accuracy

Figure 13 presents the experimental results.

Figure 13 shows the measurement results for the search accuracy of the proposed method. The horizontal axis is the number of clusters, and the vertical axis the search accuracy. The search accuracy was evaluated according to the precision when the search results for TAS were assumed to be correct answers.

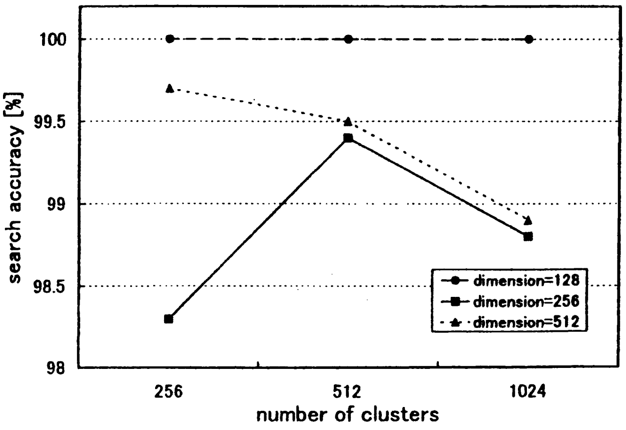


Fig. 13. Relationship between the number of clusters and search accuracy.

As shown in Fig. 13, a high search accuracy of at least 98% was obtained for every number of dimensions. In particular, when the number of dimensions was 128, a 100% search accuracy was obtained for every number of clusters. In other words, the same search accuracy was obtained as for TAS.

As an example that demonstrated good performance in these results, Table 1 shows measurement results when the number of histogram dimensions $L = 128$, number of clusters $C = 1024$, and radius parameter $p = 1$.

4.3. Video signals

To show that the proposed method can also be applied to video signals, we investigated the time required to search for a specific 15-second video signal from 24 hours of video signal data and the search accuracy.

First, we played a 24-hour tape (VHS HiFi, triple mode) that was recorded on a home VCR to capture the video on a workstation. In addition to a one-time capture of 24 hours for use as the stored signal, we randomly selected and played 10 different 15-second signals from the same tape and captured them separately for use as the reference signal. In both cases, we captured the video signal using a 29.97-Hz frame rate, Motion JPEG, and a 320 × 240 screen size. We performed feature extraction from the captured video signals by using the same method as was used in Ref. 10. We set $W = 6$ (3 equal divisions in the horizontal direction and 2 equal subdivisions in the vertical direction) as the number of subdivisions for feature extraction.

For the number of histogram dimensions, we set $L = 128$, and for the number of clusters, we set $C = 1024$. Also, for the search threshold value θ_1, we set $\theta_1 = 0.8$ so that both the precision and the recall when TAS was used would be 100%. For the radius parameter, we set $p = 1$.

Table 2 shows measurement results for the search execution time and match frequency. The average value when measurements were performed for the 10 reference signals are shown for the search execution time and match

Table 1. Experimental results for a 200-hour stored audio signal (summary)

Search execution time		Search time reduction rate
Proposed method	TAS method	
306 ms	2847 ms	9.3
Match frequency		Match frequency reduction rate
Proposed method	TAS method	
13,256 times	112,120 times	8.5

Table 2. Experimental results for a 24-hour stored video signal (summary)

Search execution time		Search time reduction rate
Proposed method	TAS method	
4.2 ms	90.0 ms	21.4
Match frequency		Match frequency reduction rate
Proposed method	TAS method	
1,153 times	8,597 times	7.5

frequency. The experiments in this section verified that the search results were correct for all 10 reference signals. These results demonstrated that the proposed method is also effective for video signals.

5. Conclusions

In this paper, we proposed a search method that introduces global pruning in addition to the local pruning used in the Time-series Active Search method. We introduced the radius parameter p to determine the selection threshold value from the search threshold value during global pruning and showed experimentally that when p is increased, the search time and match frequency are reduced while maintaining the search accuracy. We also showed experimentally that when the number of clusters is increased, the search execution time is reduced while maintaining the search accuracy. In particular, when the number of histogram dimensions was set to 128 and the number of clusters was set to 1024, the search time was reduced to approximately 1/9 of the time for the conventional method when using $p = 1$. In addition, we showed that the proposed method, like the Time-series Active Search method, can be applied to video signals as well as audio signals. For video signals, when the number of histogram dimensions was set to 256 and the number of clusters was set to 1024, the search time was reduced to approximately 1/20 of the time for the conventional method when using $p = 1$.

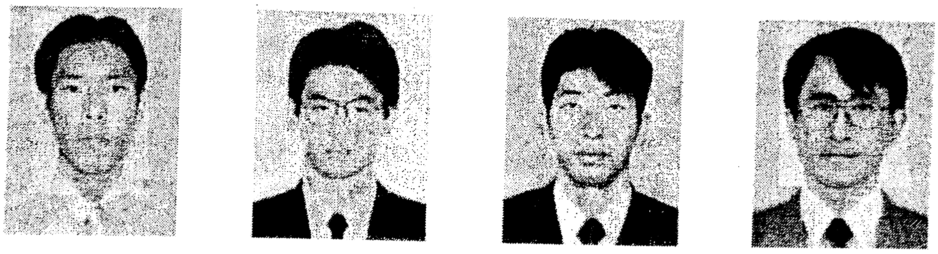
In the future, we plan to continue investigating techniques that will enable even quicker searches of more massive time-series signals while preserving search accuracy.

Acknowledgments. The authors thank Drs. Kenichiro Ishii and Noboru Sugamura of NTT Communication Science Laboratories for their ongoing guidance. They also thank everyone in the Media Recognition Research Group of NTT Communication Science Laboratories for constant cooperation.

REFERENCES

1. Dharanipragada S, Roukos S. A fast vocabulary independent algorithm for spotting words in speech. Proc ICASSP98 1998;1:233–236.
2. James DA, Young SJ. A fast lattice-based approach to vocabulary independent word spotting. Proc ICASSP94 1994;1:377–380.
3. Kosugi N, Nishihara Y, Konya S, Yamamuro M, Kushima K. Music retrieval system using Hamming. Tech Rep IEICE 1999;DE99-18.
4. Wactlar HD. Informedia—Search and summarization in the video medium. Proc Imagina 2000 Conference, 2000.
5. Mohan R. Video sequence matching. Proc ICASSP98 1998;6:3697–3700.
6. Kashino K, Smith G, Murase H. Time-series active search for quick retrieval of audio and video. Proc ICASSP98, Vol. 6, p 2993–2996.
7. Beckman N, Kriegel HP. The R*-thee: An efficient and robust access method for points and rectangles. Proc ACM SIGMOD 90, p 322–331, 1990.
8. White DA, Jain R. Similarity indexing with the SS-tree. Proc ICDE96, p 516–523, 1996.
9. Katayama N, Satoh S. The SR-tree: An index structure for high-dimensional nearest neighbor queries. Proc ACM SIGMOD Conference, p 369–380, 1997.
10. Kashino K, Kurozumi T, Murase H. Quick AND/OR search for multimedia signals based on histogram features. Trans IEICE 2000;J83-D-II:2735–2744.
11. Vinod VV, Murase H. Focused color intersection with effective searching for object extraction. Pattern Recognition 1997;30(10).
12. Garsho A, Gray RM. Vector quantization and signal compression. Kluwer Academic; 1992.
13. Sugiyama M. Fast segment search algorithms. Tech Rep IEICE 1998;SP98-141.

AUTHORS (from left to right)



Akisato Kimura (member) graduated from the Department of Electrical and Electronic Engineering at Tokyo Institute of Technology in 1998, completed the master's program in 2000, and joined NTT Communication Science Laboratories of NTT Corporation. He has been engaged in research concerning pattern recognition and multimedia information retrieval. He is interested in information theory and learning theory. He is a member of the Society of Information Theory and Its Applications, the Database Society of Japan, and IEEE.

Kunio Kashino (member) graduated from the Department of Electronic Engineering at the University of Tokyo in 1990, completed the doctoral program in 1995, and joined NTT Corporation. He is currently the senior research scientist at NTT Communication Science Laboratories. He has been engaged in research concerning search, recognition, and separation of audio signals and information integration. He is interested in signal processing and knowledge processing of media information. He received the Encouragement Award from the Information Processing Society of Japan in 1993, the Kiyoshi Awaya Academic Encouragement Award from the Acoustic Society of Japan in 1999, the Society Award from IEICE in 2001, and the Achievement Award from IEICE in 2002. He is a member of the Information Processing Society of Japan, the Acoustic Society of Japan, the Japanese Society for Artificial Intelligence, and IEEE.

Takayuki Kurozumi (member) graduated from the Department of Physics at Tokyo Metropolitan University in 1997, completed the master's program at the Japan Advanced Institute of Science and Technology in 1999, and joined NTT Corporation. He is currently working for NTT Communication Science Laboratories. He is interested in research concerning pattern recognition and image processing. He received the Achievement Award from IEICE in 2002.

Hiroshi Murase (member) graduated from the Department of Electronic Engineering at Nagoya University in 1978, completed the master's program in 1980, and joined Nippon Telegraph and Telephone Public Corporation. Since then, he has been engaged in research concerning text and image recognition, computer vision, and multimedia recognition. He was a visiting researcher at Columbia University from 1992 to 1993. Since 2003, he has been a professor in the Information Science Laboratories at Nagoya University. He received the Achievement Award from IEICE in 1985, the Telecommunications System Technology Award from the Telecommunications Advancement Foundation in 1992, the IEEE-CVPR International Conference Best Paper Award in 1994, the Yamashita Memorial Research Award from the Information Processing Society of Japan in 1995, the IEEE-CVPR International Conference Best Video Award in 1996, the Takayanagi Memorial Encouragement Award from the Takayanagi Foundation for Electronics Science and Technology and the Society Award in 2001, and the Achievement Award from IEICE in 2002. He holds a Ph.D. degree in engineering, and is a member of the Information Processing Society of Japan and IEEE.