Electronics and Communications in Japan, Part 3, Vol. 86, No. 12, 2003
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J83-D-II, No. 12, December 2000, pp. 2735–2744

A Quick AND/OR Search for Multimedia Signals Based on Histogram Features

Kunio Kashino, Takayuki Kurozumi, and Hiroshi Murase

NTT Communication Science Laboratories, Atsugi, 243-0198 Japan

SUMMARY

The problem of finding the point in time at which a known audio or video source (reference signal) appears in a long audio or video source (input signal) is referred to as a time series search, in contrast to a text string search. In a time series search, a search can be performed quickly by combining the audio and video, and then identifying several search conditions using logical equations. Thus, in this paper the authors describe the application of the time series active search method, a method to search audio signals proposed in the authors' previous paper, to video searches. Next, the authors propose an efficient algorithm for AND searches and OR searches of reference signals. In addition, the authors propose a multimodal AND search which combines audio and video. The proposed algorithms are faster than combining the results of searches performed individually. For instance, in an OR search of a reference signal, when the mutual similarity in the reference signals is above 0.8, five reference signals can be searched in a search time that is 1.2 times faster than searching one reference signal. © 2003 Wiley Periodicals, Inc. Electron Comm Jpn Pt 3, 86(12): 54–64, 2003; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ecjc.10142

Key words: time series search; multimedia search; active search; quick search.

1. Introduction

In recent years, multimedia data has come to flow all around us in vast quantities. As a result, the need for retrieval technologies and search technologies for multimedia is increasing.

Here, retrieval refers to designating various conditions related to the content of the images and sounds to be found, then acquiring from a database or long-term materials specific sounds and images that meet these conditions. Retrieval is also referred to as content retrieval. There have been many reports on research related to audio and video content retrieval [1–7]. On the other hand, a search refers to finding out where specific sounds and videos (reference signals) are located in a database or lengthy materials (input signal).

The focus of this paper is on quick search technologies. In the same way that a high-speed text string search algorithm plays a vital role in handling text data, a quick time series search algorithm is very important for handling multimedia data. In practice, for instance on the Internet, high-speed time series search methods are needed to prevent illegal use of music, video, and other copyrighted materials. In addition, if a quick time series search method is available, program titles and commercials, as well as other particular sound and video broadcast dates and times can be picked up in a short period from lengthy television broadcast data.

The authors have already proposed a time series active search method, a method for quick searches of audio signals, in a previous paper [8]. However, in order to increase the usability of a time series search, (1) the audio and video should be searchable in any combination and (2) various search conditions should be usable quickly by using logic (AND/OR).

Thus, the authors propose a basic algorithm to quickly search a time series flexibly in this paper. Section 2 describes the new visual characteristics in addition to explaining in brief the time series active search method. Section 3 discusses the OR search for reference signals, and Section 4 discusses the AND search for reference signals, both used to reduce the number of references. In Section 5, the AND search for modality is discussed for when audio and video synchronized to an input signal, that is, a multimedia AND search, is used. Section 6 offers an experimental evaluation of the validity of the method discussed in earlier sections, and Section 7 presents conclusions.

2. Time Series Active Search Method

2.1. Outline of the algorithm

The most fundamental method for a time series search is to perform signal detection [9] based on the correlation of features extracted from the signal itself (audio and video). However, this method is limited in that considerable time is required when using lengthy signals, and so some form of higher-speed method is needed. If audio and video information is converted to text and symbols during voice recognition and object recognition, and then a text string search is performed during retrieval, the search can be effected quickly. Under current recognition technology, however, the precision of conversion to text and symbols is not necessarily sufficient. Although there is a method which converts the signal to a string of symbols using a vector quantization (VQ) method, under a method that performs direct references of VQ symbol pairs, considerable processing time is required, as will be explained in Section 6.

In Ref. 8, the authors proposed a method for a time series active search, a high-speed algorithm for acoustic signals. This method features the use of histogram pairs for VQ symbols. The histogram is a cumulative feature, and so is not readily affected by variations in the signal. In addition, referencing the histogram pairs involves less computation for the referencing compared to direct referencing of the feature pairs [10], and so useless referencing can be skipped by finding the areas for which referencing is not needed along the time axis. The authors have reported that as a result of these effects, search precision sufficient for practical purposes can be maintained, while search speeds several hundred times faster than methods based on internal products for spectrum feature vectors can be obtained.

Below the time-series active search method is outlined. Figure 1 shows the flow of processing. First, the feature vectors for the reference signal (the short signal used for the search key) and the input signal (long signal) are extracted. Next, vector quantization (classification) is performed for the feature vectors within the time window that is the same size for both the reference signal and the input signal, and a histogram is created by counting the number of times each quantized symbol appears. Then, whether or not the reference signal appears is determined by whether or not the similarity between the histograms exceeds a previously set value (this is called the search threshold). At this point, the time width (width that can be skipped) over which the search can be skipped in the time direction can be found from the similarity value and the set value. As a result, the window can be shifted to this extent with respect to the input signal and the search can be continued.

This algorithm does not depend on a particular feature vector or its quantization method, but rather can use various different ones. In addition, the similarity between the histograms can be defined in a variety of ways. Among these the authors have focused on the histogram weighting in particular. The weighting S_{IR} for the histograms H_I and H_R can be defined as

$$\begin{aligned} S_{IR} &= S(H_I, H_R) \\ &= \frac{1}{D}\sum_{l=1}^{L}\min(h_{Il},\ h_{Rl}) \end{aligned} \tag{1}$$

Here, H_I and H_R are histograms for the input signal and the reference signal, and h_{Il} and h_{Rl} are the numbers of feature vectors quantized by the l-th symbol. Also, L is the size of the VQ symbol width, and D is the length of the reference signal (total number of feature vectors derived from the reference signal).

At this point, the skip width w can be found using the following equation based on the upper limit for the similarity:

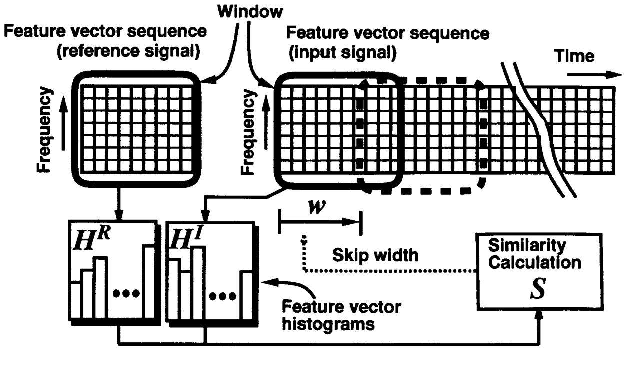
Window
Feature vector sequence (reference signal)
Feature vector sequence (input signal)
Time
Frequency
Frequency
H^R
H^I
w
Skip width
Similarity Calculation
S
Feature vector histograms

Fig. 1. Overview of the time-series active search.

$$w = \begin{cases} \text{floor } (D(\theta - S_{IR})) + 1 & (S_{IR} < \theta) \\ 1 & (\text{other than above}) \end{cases} \quad (2)$$

Note that floor (•) represents the undercut, and θ the search threshold. Where the similarity exceeds θ, a complete search is performed (the time window is shifted one whole unit).

In Eq. (1), the reason for using the similarity definition based on the histogram weighting from among all the definitions that can be conceived is: (1) the similarity calculation is simple; (2) the upper limit for the similarity in the histogram obtained by shifting the time window can be found using a simple computation; (3) it has already been used for object recognition in images, and the results have been favorable [11–13]. Note that Sugiyama has considered the time series [14] for the upper limit of the similarity and the definition of similarity in the time series active search method.

2.2. Acoustic features

The conditions required for the feature vectors in the time series active search include good discrimination performance (similarity is high in the target area, and low in nontarget areas), robustness (minimally affected by the recording conditions for sound or images, or noise), and a reasonable amount of computation for feature extraction.

In consideration of these points, the short-term power spectrum for the feature vector used in this paper is found while shifting analysis frames using a bandpass filter, and then the spectrum is normalized for the frequency channel. In other words, the acoustic feature vector $\boldsymbol{f}(k)$ can be written as

$$\boldsymbol{f}(k) = (f_1(k), f_2(k), \ldots, f_V(k)) \quad (3)$$

Here, k is the discrete time ($k = 1, 2, \ldots$) which represents the position of the analysis frame, and V is the number of dimensions for the feature. Each element of $\boldsymbol{f}(k)$ is

$$f_j(k) = \alpha(k)\ Y_j(k) \quad (4)$$

Here, $Y_j(k)$ is the average of the square of the analysis frame for the output waveform of the j-th bandpass filter. In addition, $\alpha(k)$ is the coefficient used for normalization, and is defined as

$$\alpha(k) = \frac{1}{\max\limits_{j}(Y_j(k))} \quad (5)$$

2.3. Video features

The time series active search method enables searches for video (time series pictures) by using video features [15]. In this paper, after considering the robustness with respect to the differences in characteristics in various video machines, the authors focused on brightness. In other words, the video feature vector $\boldsymbol{g}(k)$ can be written as

$$\boldsymbol{g}(k) = (g_1(k), \ldots, g_W(k)) \quad (6)$$

Here, k represents the time for the frame, and the subscript with g, the breakdown number for the W subimages the image in each frame is broken down into. g_j is the value of the brightness for each pixel averaged and normalized in a subimage:

$$g_j(k) = \frac{\bar{x}_j(k) - \min\limits_i \bar{x}_i(k)}{\max\limits_i \bar{x}_i(k) - \min\limits_i \bar{x}_i(k)} \quad (7)$$

However,

$$\bar{x}_i(k) = \mathop{\mathrm{E}}_{p \in \Omega} [x_p(k)] \quad (8)$$

Here, Ω is the set of pixels p in the i-th subimage, and E represents the average of 2. In addition, with the RGB values for the pixel p expressed as r_p, g_p, b_p,

$$x_p = 0.299r_p + 0.587g_p + 0.114b_p \quad (9)$$

This is a relational expression for RGB values and in the NTSC format.

3. OR Searches for Multiple Reference Signals

Applications for a time series search might include counting the number of times a particular commercial or song appears in a data which stores television broadcasts or radio transmissions, or a search engine for acoustic signals on the Internet. Such uses frequently include performing searches with respect to a large number of reference signals (OR searches). For instance, when performing a count of the commercials in broadcast data, even if the focus is on commercials for the same product, ordinarily several similar commercials which are slightly different are broadcast over the same time period. In addition, for the number of times a song is used as well, often people want to search for several instances of several songs at the same time. Thus, the purpose of this section is to explain a method for reducing the amount of reference computation even when repeating the same search when performing an OR search on multiple reference signals with respect to one input signal.

Let us assume that the histograms H_{Rj} are created from the N reference signals R_j ($j = 1, 2, \ldots, N$), then the histogram H_I is created from the position of the current time window for the input signal I. Note, however, that we will assume that the total count D_j for H_{Rj} is always equal, and that $D_j = D$. Now, for H_{Rm} and H_I for which $j = m$, the similarity S_{IRm} is obtained using the calculations in Eq. (1). The authors are interested in finding the upper limit for S_{IRj} making specific reference to H_I and H_{Rj} ($j \neq m$). Thus, when considering the upper limit for S_{IRj},

$$S_{IRj} \leqq 1 - [S_{IRm} - S_{RmRj}] \tag{10}$$

can be shown to be valid (see the Appendix).

Given this, the OR search for multiple reference signals can be performed as shown below.

(1) The similarity for reference signal pairs is calculated for all combinations as a preprocess.

(2) The current position is set to the start of the input signal. (Search process begins here.)

(3) The reference signal for which the skip position is closest to the current position is selected, and the current position is set to that skip position.

(4) The selected reference signal and the input signal at the current position are compared, and the similarity is found.

(5) Based on the resulting similarity, the skip widths for all of the reference signals are updated.

(6) Return to (3).

As a result, the number of instances of referencing during the search process can be reduced to below when reference signals are referenced individually. Note that when D_j is not equivalent, if D is used as the minimum for D_j, then the above argument will hold for the subspace of length D.

4. AND Searches for Multiple Reference Signals

An AND search for multiple reference signals represents finding the interval with a similarity that exceeds the search threshold value for any of $R_j (j = 1, \ldots, N)$ among the input signals when multiple reference signals $R_1, \ldots, R_N$ are positioned on the time axis with the time delays $\tau_1, \ldots, \tau_N$ since the start time. The length t_d of the interval being searched is

$$t_d = \max_j(\tau_j + D_j) - \min_j(\tau_j) + 1 \tag{11}$$

Here, D_j represents the continuous time for R_j. If the overall similarity S for the time interval t_d is defined as follows, the problem becomes searching for places where S exceeds the search threshold:

$$S = \min_j(S_j) \tag{12}$$

Here, S_j is the j-th reference signal, and is the similarity with the interval which corresponds to the input signal.

The above is also important in an AND search of reference signals as well. In the time series active search method, although affecting small changes in the signals is difficult due to referencing with the accumulated features, the time structure is stored as a time series, and as a result, distinguishing the differences in the order of appearance tends to be difficult. In such cases, if the reference signals can be broken down and an AND search performed on each block, then accurate distinctions can be made.

The basic approach for an AND search is to perform a search in order for each reference signal. In this paper this approach will be referred to as the sequential method. First, the j-th skip width w_j for the current position in the total time window (length t_d) is found. Then, the skip width w for the total time window

$$w = \max_j(w_j) \tag{13}$$

can be found. Based on Eq. (12), if any of S_j is below θ, then S must also be below θ. As a result, S cannot exceed θ even if the time window is moved as much as possible for j in w_j. In practice, skipping the total time window immediately when S_j which does not satisfy the search threshold value θ usually allows for fewer reference computations as compared with referencing every j. In this paper this approach shall be referred to as the sequential interrupt approach.

As an aside, when performing AND searches for several reference signals broken down in terms of time from one original reference signal (in other words, when several reference signals that are adjacent to each other in terms of time have an AND search performed on them),

$$S_A \geqq \min_j(S_j) \tag{14}$$

can be established (see the Appendix). Here, S_A is the similarity for the original reference signal, and S_j is the similarity for the broken-down reference signals. This means that the part for which the similarity exceeds the threshold value for the AND search is not skipped ever, even during a search of the original reference signal. As a result, an independent search of the original reference signal should be performed first, and then only the necessary portions should be compared for each reference signal. This will be referred to as the merged method in this paper.

When combining the same reference signal and input signal, both the merged method and the sequential method (or the sequential interrupt method) will on average have few reference computations. For instance, if a reference signal is broken down into N equal parts, the conditions

$$w_A \geqq \max_j(w_j) \tag{15}$$

can be calculated using

$$S_A \leqq \theta - \frac{1}{N}(\theta - \min_j(S_j)) \tag{16}$$

with w_A as the skip width for the merged method. Equation (16) does not always hold, however. If we assume that the similarity for each segment is uniform [in other words, $\min_j (S_j) = S_A$], then Eq. (16) will hold for $S_A \leqq \theta$. Given this, in many cases the merged method will be more useful than the sequential method.

5. Multimodal AND Search

In the previous sections the authors have discussed OR searches and AND searches for several reference signals with respect to one input signal. In addition, however, in many cases people want to perform OR searches or AND searches for reference signals which correspond to several synchronous input signals. Here, for multimodal AND searches in particular, in other words for searches of audio and video signals which are synchronized, we will evaluate searching for locations where either reference signal is similar.

In a multimodal AND search, the similarity S is defined for all input signals as the minimum value among the similarity S_j for each input signal, just as was done for an AND search of several reference signals. Then, the skip width w_j found for each input signal is calculated in real time. By using the largest skip width in real time, the number of references can be lowered to below when searching each input signal separately. This is shown in Fig. 2.

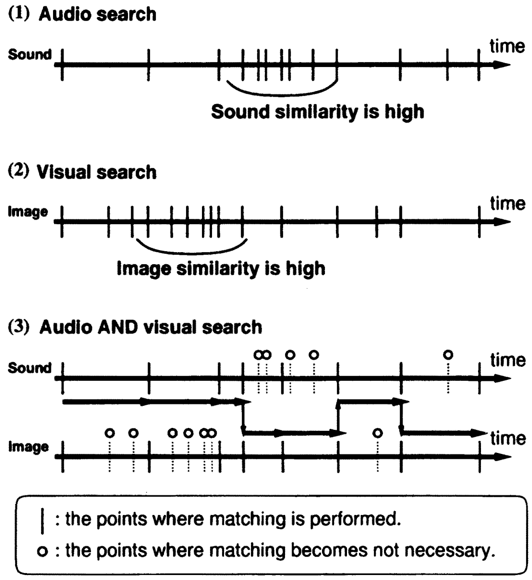
(1) Audio search
Sound
time
Sound similarity is high
(2) Visual search
Image
time
Image similarity is high
(3) Audio AND visual search
Sound
time
Image
time
| : the points where matching is performed.
o : the points where matching becomes not necessary.

Fig. 2. Skip width example in the search combining audio and video (multimodal AND search).

6. Experiments

6.1. Video features

First, experiments on search speed and search precision were performed with respect to the video characteristics described in Section 2 in order to evaluate the effectiveness of the video features. Table 1 lists the specifications of the workstation used in the experiment.

6.1.1. Search speed

In order to evaluate the search speed, the authors measured the time required to search for a particular 15-second commercial among 6 hours worth of images in a television broadcast.

First, the authors recorded 6 hours of a private television network's broadcasts using a home VCR (VHS, HiFi, 3x mode). Next, the authors imported the video to the workstation mentioned by playing back the recorded tape. In addition to importing the 6 hours for use as an input signal, 10 different 15-second commercials were selected and played back randomly from the same tape for use as a reference signal, then imported separately from the input signal. In both cases, the import was performed at a frame rate of 29.97 Hz with uncompressed RGB and a screen size of 80 × 60. Two bins were used for each dimension in the feature vector. Also, W was set to 12 (divided into 4 for the horizontal and 3 for the vertical).

The time required for the search consists of (1) the time required to extract features (the feature extraction time), (2) the time required for vector quantization of the feature vector (vector quantization time), and (3) the time required to execute the search using the results of vector quantization (the search execution time, in other words the time required to create the histogram, calculate the similarity, and repeat the window movements). Note that the time

Table 1. Specifications of the workstation used in the experiments

Name of model	SGI O_2
CPU	R10000 (250 MHz)
Memory	384 MB
OS	IRIX Release 6.3
Compiler	MIPSPRO C Compiler ver. 7.00

Table 2. Search speed based on the image features

Search execution time		Speed improvement	Ratio of number of references	Reference
Total search	Proposed method			
22.5 s	0.20 s	112 times	1/207	Figure 3

Total search refers to a search in which the skip width is set to 1.

in the discussion below was measured using CPU time. CPU time appears to vary by several percent with each measurement, and so below the average value for measurements taken five times is used.

For (1) the feature extraction time, the CPU time required to calculate the features from a 6-hour input signal and 15-second reference signal was about 650 seconds. In other words, feature extraction can be performed in about 3% of real time. Therefore, if processing is performed at the same time as the signal is brought into the computer, then features can be extracted at a 3% CPU load.

For (2) the vector quantization time, the CPU time required for vector quantization of the 6-hour and 15-minute feature vector was about 0.86 second. This represents a measurement of the time for processing using memory after all the feature vectors have been loaded into memory.*

For (3) the search execution time, Table 2 shows the results of the measurements. The search execution time depends on the reference signal, the input signal, and the search threshold value.

The CPU time shown in Table 2 is the average value of measurements taken five times for the ten reference signals (commercials). In this experiment, the threshold value was set to $\theta = 0.6$. Table 2 shows that the number of references in the proposed method was lowered on average compared to when a total search was used (ratio of number of references). Note that in this experiment, the authors confirmed that the search results for all of the ten commercials were correct (no search leaks or extra searches, and a time difference in the search results of less than 1 second).

For reference, Fig. 3 shows an example of the pattern of changes of time of the similarity in this experiment (when using a commercial as the reference signal). The white circles represent time points for a search, and the dotted line represents the search threshold.

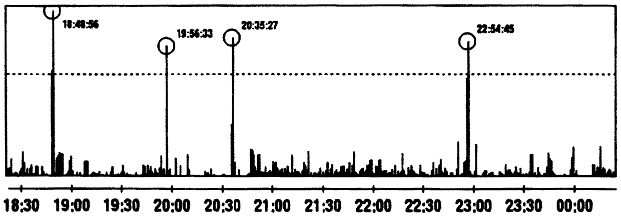

Fig. 3. Search result by the image feature. The horizontal axis shows the time, and the vertical axis shows the similarity (0–1).

Note that a method for a direct comparison with the VQ signal seen as a character string would be to count in sequence the rate of corresponding VQ signal pair matches. The search execution time in this instance was about 2.4 seconds using the settings in this experiment. This is about 12 times greater than for the proposed method.

6.1.2. Search precision

In order to investigate the search precision of the proposed method, the authors performed an experiment using a separate recording of a television broadcast. First, the authors recorded a television broadcast using the same method as was used for the first experiment. They edited it to link together distinct commercials into a 1-hour piece. This was done because using materials in which the same image does not appear twice was better for automating the experiment. This video tape was played back, and the images were divided and imported into a workstation. One portion was used as the reference signal (it was segmented into set time lengths at random locations), and the other was used as the input signal for searches. An experiment was also conducted for when black-and-white Gaussian noise was added for the RGB values in the input signal.

In this experiment, the length and SN ratio for the reference signal were used as parameters. The SN ratio was controlled by setting the noise distribution with respect to the average of the square of the RGB values for the 1-hour input signal. A search was performed 200 times under the same experimental conditions, and the precision was measured. Precision was evaluated using the average of the precision rate and the recall rate. Here, the precision rate is the rate of correct results among the search results output, and the recall rate is the rate of search results output in what was supposed to be output. The precision rate and the recall rate vary depending on the search threshold set. In this experiment, the search threshold was varied by controlling c in the following expression:

$$\theta = m + c\,\sigma \tag{17}$$

*In the setup in this paper, one 12-dimensional feature vector is stored in 12 bytes of memory, and as a result the feature vector for 6 hours of input takes up 7.8 MB of memory net.

Here, *m* and σ are the average and the standard deviation for the similarity gathered by sampling the input signal for a given reference signal and then calculating the similarity beforehand. Equation (17) was found experimentally during preparatory experiments. Also, in Eq. (17), when θ exceeds 0.9 it is set to 0.9, and when it is less than 0.1, it is set to 0.1. In this experiment, the value of *c* in Eq. (17) was set during 200 iterations, and by adjusting this steady value, a value which maximizes the precision became the evaluation value.

Equation (17) represents setting θ to different values in consideration of the distribution of similarity for individual reference signals. Note that a specific value for *c* in this experiment ran in a range from 4.8 to 23, and as the noise power grew, the value of *c* which maximized the precision tended to be smaller.

The other input and search parameters were the same as found in the experiment in the previous section.

Figure 4 shows the results of the experiment. When the reference signal was 15 seconds, no search leaks or extra searches occurred while the SN ratio was 2 dB or lower. In addition, when the SN ratio was 30 dB or higher, even when the reference signal was 2 seconds, a search precision of above 99% was obtained.

Based on these experiments, it is clear that good search precision and a search speed similar to that found in searches of acoustic signals can be obtained for video. Note, however, that the noise seen in real-world video sets frequently is far removed from black-and-white noise. Consequently, the results of the experiment on search precision described above cannot be interpreted as is for tolerance for noise in a real-world video.

6.2. OR search of multiple reference signals

The authors performed an experiment in order to explore to what extent the proposed method increases the speed of an OR search compared to searching multiple reference signals individually, The algorithm used for the OR search and the AND search described in this paper can be used in the same way when searching for video or audio. Here an example of an audio search is given. In other words, in this experiment, the audio signal from 6 hours' worth of television broadcasts is used as the input signal, and five 15-second signals are used as the reference signal. Because the proposed method is the same as when comparing reference signals individually in terms of precision, the search speed (search execution time) and the number of comparisons are compared.

Given the discussion in Section 3, the extent to which an OR search can be made more efficient by the proposed method depends on the precision among the reference signals (mutual similarity). Thus, five reference signals are created by connecting the shared acoustic signal (part of a particular commercial) and the acoustic signal that is not shared, then the mutual precision is regulated by controlling the length of the shared part. In addition, the input signal is a signal that does not include the acoustic signal used to create the reference signal.

The parameters for the search are: sampling frequency = 11.025 kHz, number of feature dimensions = 7, length of analysis frame = 60 ms, shift width of analysis frame = 10 ms, number of bins in each feature dimension = 3, search threshold θ = 0.8.

Figure 5 shows the results of the experiment. The thick dashed lines represent the total search execution time and the total number of comparisons when searching five reference signals individually, and the thin dashed line represents 1/5 of that. In Fig. 5, the number of comparisons and the search execution time are both values from the five

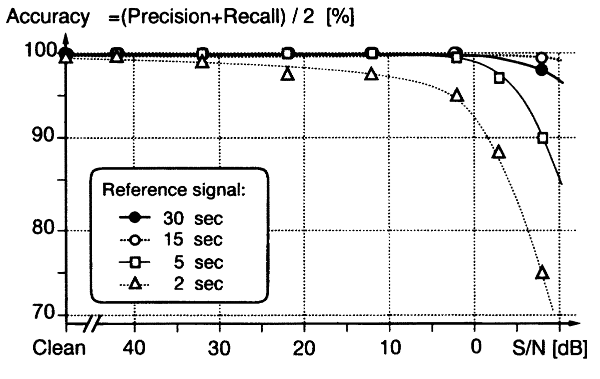

Fig. 4. Search accuracy based on the image features.

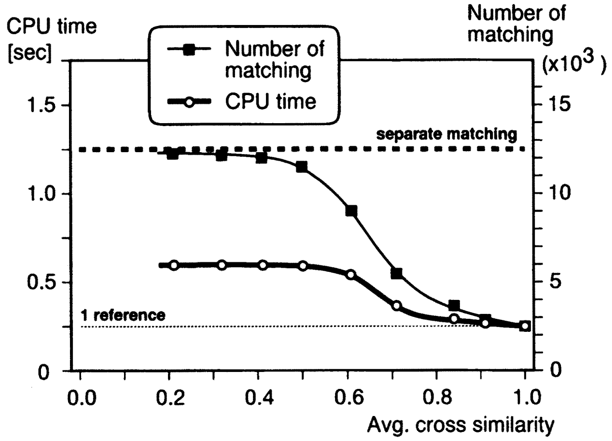

Fig. 5. Number of matches and CPU time in the OR search with 5 reference signals.

reference signals, including the prior comparisons among the reference signals. As is illustrated in Fig. 5, when the average mutual similarity is close to 1, the search can be performed with a number of comparisons and in a search execution time that approaches a search of one reference signal. For instance, when the average mutual similarity is 0.84, the number of comparisons is 1.5 times greater and the search execution time is 1.1 times greater than that when one reference signal is searched. In addition, when the mutual similarity falls and the number of comparisons approaches that of individual searches, the search execution time falls to 0.47 times that of an individual search. This is thought to be due to the cost of creating a histogram for the input signal being lower for an OR search compared to the total in an individual search.

In the proposed method, the number of comparisons necessary to compare reference pairs is $N(N - 1)/2$ for N reference signals. When there are five reference signals, as was the case in this experiment, the number of comparisons was 10, sufficiently low to be ignored. Note, however, that as N rises, the number of comparisons rises on the order of the square of N. For instance, when there are 1000 reference signals, roughly 500,000 comparisons will be required to compute the similarity among reference signals. In an individual search, about 2500 comparisons are needed to compare one reference signal, and as a result about 250,000 are needed for 1000. When the mutual similarity between reference signals is low (for instance, 0.2), a total of about 3 million comparisons are estimated to be necessary in the proposed method.

6.3. AND search of multiple reference signals

Using experiments the authors confirmed the number of comparisons and the search speed (search execution time) when performing AND searches of multiple reference signals. As was the case for the video search experiment in Section 6.1, the video signal from 6 hours of television broadcasts was used as the input signal, and 15 seconds of 10 commercials selected at random were used for the search. Each commercial was broken into three equivalent lengths so as to avoid duplication and used as three reference signals. AND searches were then performed. The search parameters were the same as in Section 6.1.

Table 3 shows the results of measurements for the number of matches and the search execution time. For all cases, (a) sequential method, (b) sequential interrupt method, and (c) merged method, the number of matches was reduced compared to when performing an individual search for three reference signals. Note that the search results for (a), (b),and (c) were the same [when performing the searches individually, because the reference signals were different, the results of (a) to (c) cannot be compared]. The reduction in the number of matches in (a) compared to an individual search is the result of a maximum in Eq. (13) being used. In this experiment, the effect of (c) (the merged method) was most significant among the three methods (a) to (c).

Table 3. The number of matches and CPU time in the AND search with respect to the reference signals

Type of search	Number of matches ($\times 10^3$)	Search execution time
Individual	22.49	1.40 s
(a) AND (sequential method)	21.86	1.43 s
(b) AND (sequential interrupt method)	7.52	0.34 s
(c) AND (merged method)	2.63	0.20 s

On the other hand, Fig. 6 shows the similarity for when dividing a commercial into three parts and performing an AND search. This clearly showed that compared to the case in Fig. 3 of searching all 15-second commercials as one reference signal, the similarity in sites that were not correct was much lower in Fig. 6, and the margin in the similarity settings was much higher.

6.4. Multimodal AND search

The authors measured the number of matches and the search speed (search execution time) for a multimodal AND search. As in the experiments in the previous sections, the audio and video signal from 6 hours' worth of television broadcasts was used as an input signal, and the video and audio signal from 10 commercials 15 seconds long was used as the reference signal. Note that in a typical commercial search, searching for the audio or video alone would be sufficient, and so performing a multimodal AND search had

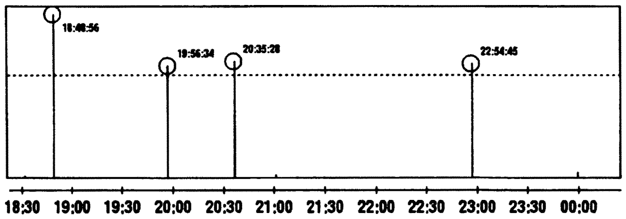

Fig. 6. Search results of the AND search with three reference signals.

Table 4. The number of matches and CPU time in the search combining audio and video

Type of search	Number of matches ($\times 10^3$)	Search execution time
Audio	11.39	0.45 s
Video	3.39	0.20 s
AND	3.04	0.18 s

little meaning. Rather, the multimodal AND search seems likely to be used when trying to narrow search results using both audio and video when a large number of search result candidates have been found for audio or video alone. Thus, in this experiment the search threshold θ was set low at 0.5 s as to yield more search results in a single modality. Note that in this experiment, when a skip width greater than a set level was found for either the audio or the video (set to 1.5 seconds based on preparatory experiments), the match in the other modality at the same time window was omitted.

The results of the experiment are shown in Table 4. It is clear that as a result of the proposed method, the number of matches and the search execution time were both reduced compared to either modality alone. Compared to the sum of performing searches individually, the number of matches for the AND search was about 21%, and the search execution time was about 28% lower.

7. Conclusion

In this paper the authors proposed a method to execute an AND search, OR search, and multimodal AND search of multiple reference signals efficiently, in addition to describing how to apply the time series active search method to an audio-visual search. In the OR search of the reference signal, the authors showed that by searching reference signal pairs beforehand, the number of comparison matches when searching for similar reference signals could be reduced compared to performing searches individually. In addition, when the search speed was increased, and especially when the average mutual similarity between reference signals was above 0.8, an OR search for five reference signals could be performed in a search time less than 1.2 times that of a search for one reference signal. Also, in AND searches of reference signals, the authors showed experimentally that a search could be performed efficiently by first merging adjacent segments and searching them.

The AND search and the OR search being evaluated in this paper are thought to be fundamental for the creation of flexible, convenient searches of multimedia data among the text search tools in the UNIX environment, for instance the grep family of tools. In the future the authors plan to move forward with analyses of search methods that allow deformation and variation to signals while maintaining the characteristics of a high-speed search.

Acknowledgments. The authors thank Kenichiro Ishii, director, and Norihiro Hagita, department head, at NTT Communication Science Laboratories for their daily guidance. They also thank the Media Recognition Research Group there for their daily cooperation.

REFERENCES

1. Wu JK, Narasimhalu AD, Mehtre BM, Lam CP, Gao YJ. CORE: A content-based retrieval engine for multimedia information systems. ACM Multimedia Systems 1995;3:25–41.
2. Flickner M, Sawhney H, Niblack W, Ashley J, Huang Q, Dom B, Yonkani M, Hafner J, Lee D, Petkovic D, Stede D, Yanker P. Query by image and video content: The QBIC system. IEEE Comput 1995;28:23–32.
3. Wold E, Blum T, Keislar D, Wheaton J. Content-based classification, search, and retrieval of audio. IEEE Multimedia 1996;3:27–36.
4. Young SJ, Brown MG, Foote JT, Jones GJF, Jones KS. Acoustic indexing for multimedia retrieval and browsing. Proc ICASSP-97, Vol. 1, p 199–202.
5. Pfeiffer S, Fischer S, Effelsberg W. Automatic audio content analysis. Proc ACM Multimedia, p 21–30, 1996.
6. Saunders J. Real-time discrimination of broadcast speech/music. Proc ICASSP-96, Vol. 2, p 993–996.
7. Subramanya SR, Simha R, Narahari B, Youssef A. Transform-based indexing of audio data for multimedia databases. Proc IEEE Conf on Multimedia Computing and Systems, p 211–218, 1997.
8. Kashiwano N, Smith G, Murase Y. A time-based active search method: a high-speed search method for acoustic signals using histogram features. Trans IEICE 1999;J82-D-II:1365–1373.
9. Hancock JC, Wintz PA. Signal detection theory. McGraw–Hill; 1966.
10. Sawai H, Yoneyama M, Nakagawa S. Several evaluations of increasing the speed of large-vocabulary voice recognition. J Acoust Soc Japan 1987;43:858–867.
11. Swain MJ, Ballard DH. Color-indexing. Int J Computer Vision 1991;7:11–32.
12. Vinod VV, Murase H. Focused color intersection with efficient searching for object extraction. Pattern Recogn 1997;30:1787–1797.

13. Murase Y, Vinod VV. An active search method: High-speed physical searches using focused color information. Trans IEICE 1998;J81-D-II:2035–2042.
14. Sugiyama Y. A segment high-speed search method. Tech Rep IEICE 1999;SP98-141.
15. Kashino K, Smith G, Murase H. Time-series active search for quick retrieval of audio and video. Proc ICASSP-99, Vol. 6, p 2993–2996.
16. Takagi M, Shimoda Y (editors). Image analysis handbook. Tokyo University Publishing; 1991. p 103.

APPENDIX

1. Proof for Eq. (10)

Figure A.1 shows the relationship between similarities. At present, S_{IRm} has just been obtained by matching the m-th reference signal and input signal. Also, S_{RmRj} is calculated and determined beforehand.

(i) When $S_{IRm} \leqq S_{RmRj}$

This is easy to understand if we consider a case in which the m-th reference signal and the j-th reference signal are very similar.

Among the elements of H_{Rm} (feature vector included in H_{Rm}), the set which contributes to the similarity with H_I is expressed as $\{H_{Rm} \cap H_I\}$, with the number of elements being $|H_{Rm} \cap H_I|$. As such, based on Eq. (1),

$$|H_{Rm} \cap H_I| = DS_{IRm} \tag{A.1}$$

In the same fashion,

$$|H_{Rj} \cap H_I| = DS_{IRj} \tag{A.2}$$

Now given that $S_{IRm} \leqq S_{RmRj}$, an inclusive relationship can be established as shown in Fig. A.2. The left side of Fig. A.2 represents the inclusive relationship between H_{Rm} and H_I defined by the matching. Here, when considering when the inclusive relationship for H_{Rj} results in $|H_{Rj} \cap H_I|$ being a maximum, the conditions (1) the elements of $\{H_{Rm} \cap H_I\}$ are all included in H_{Rj}, and (2) among the elements of H_{Rj}, the elements not in $\{H_{Rm} \cap H_{Rj}\}$ as previously determined by the number of elements (the hatched part of Fig. A.2) contribute to the similarity with all H_I. In other words, this occurs when H_{Rj} is in an inclusive relationship as shown by the thick lines on the right of Fig. A.2. Expressing this in an equation yields

$$|H_{Rj} \cap H_I| \leqq |H_{Rm} \cap H_I| + (D - |H_{Rm} \cap H_{Rj}|) \tag{A.3}$$

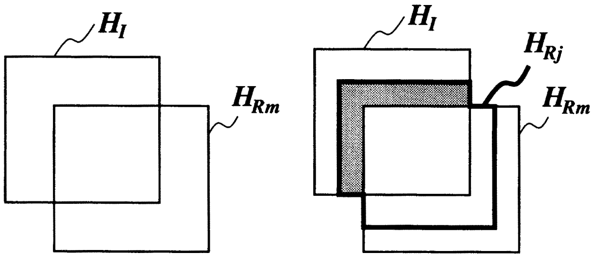

Fig. A.2. Relations between histograms (1).

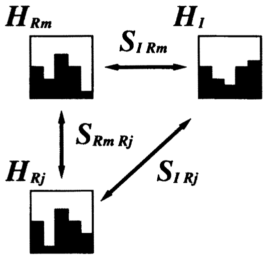

Fig. A.1. Relationship between the similarities.

Therefore, based on Eqs. (A.1) and (A.2),

$$S_{IRj} \leqq S_{IRm} + (1 - S_{RmRj}) \tag{A.4}$$

(ii) When $S_{IRm} \geqq S_{RmRj}$

This is easy to understand if we consider a case in which the m-th reference signal and the j-th reference signal are barely similar.

Using the same approach, the inclusive relationship in Fig. A.3 can be established. Therefore, the conditions in which $|H_{Rj} \cap H_I|$ reaches a maximum are (1) the elements of $\{H_{Rj} \cap H_{Rm}\}$ all being included in H_I, and (2) the elements that are not in $\{H_{Rm} \cap H_I\}$ among the elements in H_I (hatched part of Fig. A.3) all contributing to the similarity with H_{Rj}. In other words,

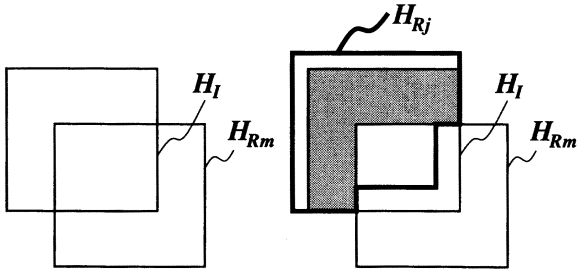

Fig. A.3. Relations between histograms (2).

$$|H_{Rj} \cap H_I| \leqq |H_{Rm} \cap H_{Rj}| + (D - |H_{Rm} \cap H_I|) \quad \text{(A.5)}$$

holds. Therefore,

$$S_{IRj} \leqq S_{RmRj} + (1 - S_{IRm}) \quad \text{(A.6)}$$

Equation (10) results when Eqs. (A.4) and (A.6) are brought together. (End)

2. Proof for Eq. (14)

If the lengths of the individual broken down segments are designated $D_1, D_2, \ldots, D_N$, then in the histogram for before the breakdown and the histograms for each breakdown,

$$DS_A = D_1S_1 + D_2S_2 + \cdots + D_ND_N \quad \text{(A.7)}$$

can be established for the number of elements contributing to the similarity with the input signal. Each segment is adjacent, and as a result

$$S_A \geqq \frac{D_1S_{min} + D_2S_{min} + \cdots + D_NS_{min}}{D_1 + D_2 + \cdots + D_N} \quad \text{(A.8)}$$

is evident [note that $S_{min} = \min_j(S_j)$]. In other words,

$$S_A \geqq S_{min} \quad \text{(A.9)}$$

and Eq. (14) holds. (End)

AUTHORS (from left to right)

Kunio Kashino (member) graduated from the Department of Electrical Engineering of the University of Tokyo in 1990, completed his doctoral studies in electrical engineering in 1995, and joined NTT. At present he is the head researcher at NTT Communication Science Laboratories. He is primarily pursuing research related to the recognition, breakdown, search, and information unification of acoustic signals. He is also interested in signal processing and recognition of media information. He holds a D.Eng. degree, and is a member of the Information Processing Society of Japan, the Acoustical Society of Japan, the Japanese Society for Artificial Intelligence, the Japanese Society for Music Perception and Cognition, and IEEE.

Takayuki Kurozumi graduated from the Department of Physics of Tokyo Metropolitan University in 1997, completed the first part of his doctoral studies in information science in 1999, and joined NTT. At present he is with NTT Communication Science Laboratories. He is interested in pattern recognition and image processing.

Hiroshi Murase (member) graduated from the Department of Electrical Engineering of Nagoya University in 1978 and joined Denden Kosha (now NTT). Since then he has been pursuing research related to character and graphical recognition, computer vision, and multimedia recognition. He was a researcher at Columbia University in 1992. At present he is a group leader in Media Recognition Research at NTT Communication Science Laboratories. He received the 1985 Scholarship Prize, the 1992 Telecom Systems Technology Award from the Electronic Communications Diffusion Corporation, the 1994 Research Paper Excellence Award from the IEEE-CVPR International Conference, the 1995 Yamashita Commemorative Research Prize from the Information Processing Society of Japan, and the 1996 IEEE-ICRA International Conference Excellence in Video Award. He holds a D.Eng. degree, and is a member of the Information Processing Society of Japan and IEEE.