# A METHOD FOR ROBUST AND QUICK VIDEO SEARCHING USING PROBABILISTIC DITHER-VOTING

*Takayuki Kurozumi, Kunio Kashino and Hiroshi Murase*

NTT Communication Science Laboratories
3-1, Morinosato Wakamiya, Atsugi-shi, 243-0198, JAPAN
E-mail:{kurozumi, kunio, murase}@eye.brl.ntt.co.jp

## ABSTRACT

We propose a quick and accurate search method for detecting a query signal from long video recordings. The method is based on the time-series active search, which is a quick searching method for audio and video signals that we previously proposed. Time-series active search is based on a histogram matching scheme and an efficient pruning mechanism, and therefore, it was very quick. We found, however, that the accuracy sometimes deteriorates when it is applied to searches through long video archives that are composed of many similar video images or those containing feature distortions caused by video dubbing or low-bit-rate compression. The problem arises from (1) insufficient capability of representing features and (2) feature distortions. Thus, the method proposed here uses LBG-based VQ to improve the capacity to represent features and Probabilistic Dither-voting to improve robustness with respect to feature distortions. The experiments prove the effects of the proposed method.

## 1. INTRODUCTION

This paper discusses a method for quickly searching long video recordings containing various distortions. The method is based on the time-series active search(TAS), which is a quick audio and video searching method that we previously proposed [1]. Once the features are extracted, TAS takes less than 1 second to detect and locate a 15-second audio or video segment in a 24-hour recording on a standard PC, under the assumption that the signal segments to be detected preserve exactly the same pattern as the stored signal, except for minor distortions or noise. However, when the method is applied to searches through long video archives that are composed of many similar video images or contain feature distortions caused by video dubbing or low-bit-rate compression, the accuracy sometimes deteriorates. The problem is here decomposed into two factors: insufficient capability of representing features in the feature space and feature distortions. We propose a method that uses LBG-based VQ to improve the capacity to represent features and Probabilistic Dither-voting to improve robustness with respect to the feature distortions. The method is expected to be applied for video retrieval from unlabeled archives, broadcasts, or the Internet [2, 3].

This paper is organized as follows: Section 2 overviews the search algorithm. Section 3 evaluates the search accuracy using a recording of real TV broadcasting. Finally, Section 4 gives conclusions.

## 2. OUR SEARCH METHOD

### 2.1. Time-series active search

Fig. 1 outlines the search algorithm. In the preparation stage, the feature vectors are calculated from both the query signal and the stored signal. The feature vectors are then quantized using a VQ algorithm. In the search stage, the windows are applied to both the query feature vectors and the stored feature vectors. Next, histograms, one for the query signal and one for the stored signal, are created by counting the number of the feature vectors over the window for each VQ code. The similarity between these histograms is then calculated. When the similarity exceeds a threshold value, the query signal is detected and located. In the last step, the window on the stored signal is shifted forward in time and the search proceeds. Here, histogram intersection is used as the similarity measure [4], and is defined as

$$S = \frac{1}{D}\sum_{l=1}^{L}\min(h_{Ql}, h_{Sl}), \tag{1}$$

where $h_{Ql}, h_{Sl}$ are the numbers of feature vectors contained in the $l$-th bin of the histograms for the query and the stored signal, respectively, $L$ is the number of histogram bins, and $D$ is the total number of feature vectors contained in the histogram. The skip width $w$ is given by

$$w = \begin{cases} \lfloor D(\theta - S) \rfloor + 1 & \text{if } S < \theta \\ 1 & \text{otherwise,} \end{cases} \tag{2}$$

0-7803-6725-1/01/$10.00 ©2001 IEEE

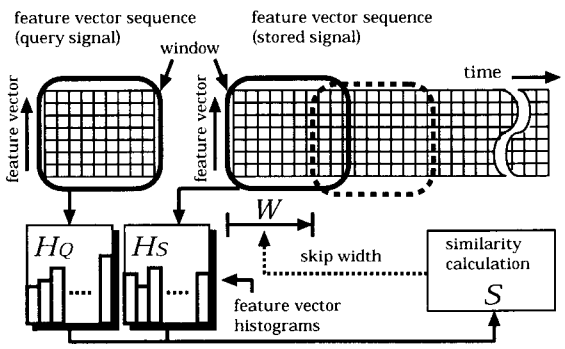


**Fig. 1**. Overview of time-series active search.

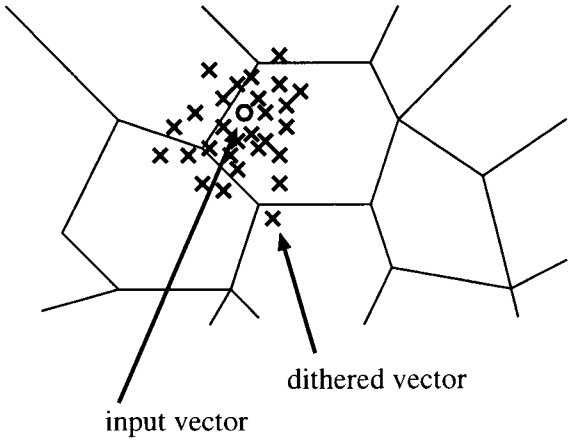


**Fig. 2**. An example of dithered vectors based on the probabilistic distribution.

where $\lfloor x \rfloor$ means the greatest integral value less than $x$, and $\theta$ is a given threshold.

### 2.2. Image Features

We use small scaled images as video features. An image feature vector is defined as

$$\begin{aligned} \boldsymbol{x}(k) &= (x_{1r}(k), x_{1g}(k), x_{1b}(k), \cdots, x_{jc}(k), \\ &\quad \cdots, x_{Wr}(k), x_{Wg}(k), x_{Wb}(k)), \end{aligned} \tag{3}$$

where $k$ is the frame number, $j$ is the division number of the subimages, and $W$ is the number of subimages. The $c$ stands for either red, green or blue. The $x_{jc}$ is the normalized intensity and is defined as

$$x_{jc}(k) = \frac{\overline{y}_{jc}(k) - \min_i \overline{y}_{ic}(k)}{\max_i \overline{y}_{ic}(k) - \min_i \overline{y}_{ic}(k),} \tag{4}$$

$$\overline{y}_{ic}(k) = \frac{1}{|I|} \sum_{p \in I} y_{pc}(k), \tag{5}$$

where $I$ is a set of pixels $p$ in the $i$-th subsection, $|I|$ is the number of pixels in $I$, and $y_{pc}(k)$ is a intensity of one color $c$ in RGB of pixel $p$.

### 2.3. Improvement of feature representation

Image feature vectors are not uniquely distributed in feature vector space. Especially, it is known that RGB intensities are highly correlated. Previously, we quantized feature vectors using combinations of scalar quantization (SQ) for each feature dimension [5]. However, the code assignment should reflect the density of the feature vectors in the feature space. That is, the regions where the feature vectors frequently occur should be assigned more quantization codes in comparison with the regions where the feature vectors infrequently occur. In this paper, we use the LBG algorithm as VQ codebook learning based on feature distribution [6].

### 2.4. Robust matching in terms of feature distortions

#### *2.4.1. Probabilistic Dither-voting*

Probabilistic Dither-voting is a method for representing a probabilistic feature distribution on the histograms. As shown in Fig. 2, a feature vector is first shifted according to probabilistic distributions. Then, those possibly multiple feature vectors are voted on. Many vectors can represent probabilistic distribution precisely, but in terms of search speed, a smaller number of vectors is preferable.

#### *2.4.2. Learning the probabilistic distribution*

In this paper, the probabilistic distribution is simply assumed to be normal distribution.

Let $\boldsymbol{x}(k)$, $\boldsymbol{x}_Q(k)$, and $\boldsymbol{x}_S(k)$ be a broadcasting signal, a query signal, and a stored signal. We prepare two signals, $\boldsymbol{x}_{Q1}(k)$ and $\boldsymbol{x}_{Q2}(k)$, through the system to obtain query signals to calculate the probabilistic distribution $\boldsymbol{\varepsilon}_Q$. We calculate the mean square error $\mathrm{MSE}(\boldsymbol{x}_{Q1}(k), \boldsymbol{x}_{Q2}(k))$ as

$$\begin{aligned} &\mathrm{MSE}(\boldsymbol{x}(k), \boldsymbol{y}(k)) \\ &= \mathrm{E}\{(\boldsymbol{x}(k) - \boldsymbol{y}(k)) \cdot (\boldsymbol{x}(k) - \boldsymbol{y}(k))\}, \end{aligned} \tag{6}$$

where E means the average for $k$.

When the variance of $\boldsymbol{\varepsilon}_Q$ is $\sigma_Q^2$,

$$\begin{aligned} &\mathrm{MSE}(\boldsymbol{x}_{Q1}(k), \boldsymbol{x}_{Q2}(k)) \\ &= \mathrm{E}\{\boldsymbol{\varepsilon}_{Q1}(k)^2\} \end{aligned}$$

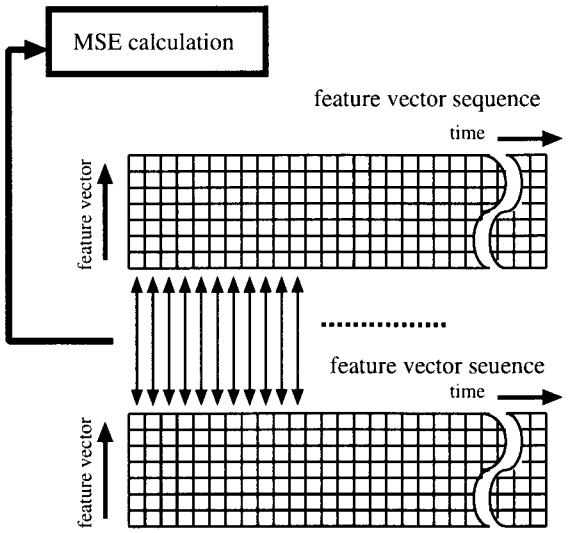


**Fig. 3**. Learning the probabilistic distribution.

$$\begin{aligned} & -2\mathrm{E}\{\varepsilon_{Q1}(k)\cdot\varepsilon_{Q2}(k)\} \\ & +\mathrm{E}\{\varepsilon_{Q2}(k)^2\} \\ = \ & \mathrm{E}\{\varepsilon_{Q1}(k)^2\}+\mathrm{E}\{\varepsilon_{Q2}(k)^2\} \\ = \ & 2\sigma_Q^2. \end{aligned} \tag{7}$$

We prepare two signals, $\boldsymbol{x}_Q(k)$ and $\boldsymbol{x}_S(k)$, to calculate the probabilistic distribution. The mean square error $\mathrm{MSE}(\boldsymbol{x}_Q(k), \boldsymbol{x}_S(k))$ is calculated as

$$\begin{aligned} & \mathrm{MSE}(\boldsymbol{x}_Q(k), \boldsymbol{x}_S(k)) \\ = \ & \mathrm{E}\{\varepsilon_Q(k)^2\}+\mathrm{E}\{\varepsilon_S(k)^2\} \\ = \ & \sigma_Q^2+\sigma_S^2, \end{aligned} \tag{8}$$

where the variance of $\varepsilon_Q$ and $\varepsilon_S$ are $\sigma_Q^2$ and $\sigma_S^2$.

Eqs.(7) and (8) are calculated prior to the search stage. This is done by comparing two signals frame by frame (Fig. 3).

## 3. EXPERIMENTAL RESULTS

### 3.1. Improvement of feature representation

The proposed method was implemented on a PC and tested with regard to search accuracy. In the first experiment, we performed a search accuracy test to show the effectiveness of distribution-sensitive feature representation. We used a 1-hour recording of many different TV commercials. In the experiment, the video frame rate was 29.97 Hz, image size was 320 × 240. We captured this recording twice. One was used for a stored signal and the other was used as the source

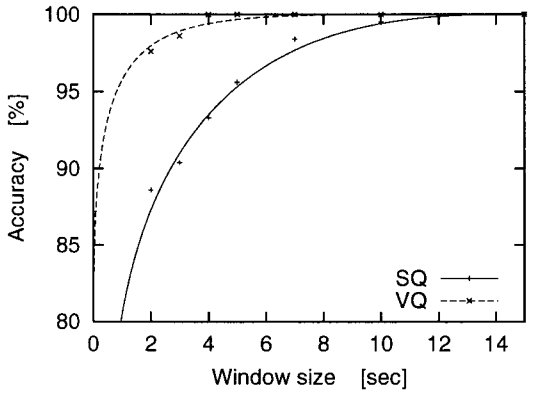


**Fig. 4**. Window size vs. search accuracy.

of the query signals: segments chosen from the second signal at random were used as query signals. The search was repeated 200 times. In the experiment, we set threshold $\theta$ using the following equation:

$$\theta = m + t\nu, \tag{9}$$

where $m$ and $\nu$ are the average and the standard deviation of similarities. The coefficient $t$ was empirically chosen. If $\theta$ is greater than 1, $\theta = 1$. If $\theta$ is smaller than 0, $\theta = 0$. In the experiment, $t$ in Eq.(9) was fixed throughout the 200 measurements. We adjusted $t$ so that the precision rate equals the recall rate. $W$ in Eq.(3) was 4, where each frame image was divided into 2 subsections both vertically and horizontally. We used a 20-minute signal that is different from the stored signal to create the codebook by the LBG algorithm. The codebook size was 4096. Fig. 4 shows the search accuracy. The LBG-based VQ (shown as "VQ") achieves higher search accuracy than SQ-based VQ (shown as "SQ"); for example, accuracy improved from 88.6% to 97.6% when the window size was 2.0 seconds.

### 3.2. Robust matching in terms of feature distortion

We performed a search accuracy test using Probabilistic Dither-voting. We used the same signal (A) as in §3.1 for the test. We used a different 20-minute signal (B) from the signal for learning probabilistic distribution.

We prepared 12 recordings,

(1) a directly captured signal,
(2) another directly captured signal,
(3) a signal dubbed twice,
(4) a signal dubbed four times,
(5) a compressed signal A,
(6) a compressed signal B

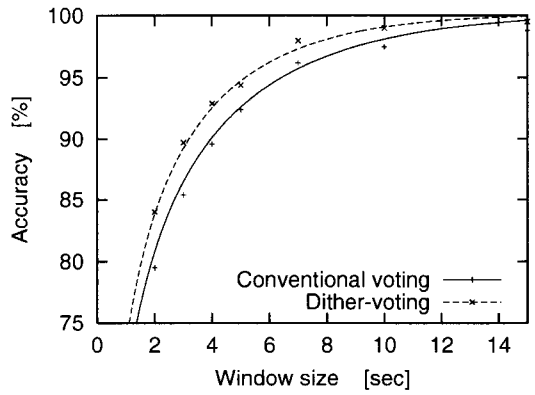


**Fig. 5**. Window size vs. search accuracy.

**Table 1**. Search accuracy for dubbed or compressed videos.

| | conventional | proposed |
|---|---|---|
| A directly captured signal | 100.0% | 100.0% |
| A signal dubbed twice | 92.3% | 95.5% |
| A signal dubbed four times | 92.4% | 94.4% |
| A compressed signal A | 99.4% | 99.5% |
| A compressed signal B | 98.8% | 99.4% |

for both (A) and (B)[1].

We performed the test by using the stored signal of a 4-time dubbed video recording (A-4). We calculated the variance $\sigma_Q^2$ by using MSE between (B-1) and (B-2) and the variance $\sigma_S^2$ by using $\sigma_Q^2$ and MSE between (B-1) and (B-4).

$W$ in Eq.(3) was 12, where the image was divided vertically into 3 subsections and horizontally into 4 subsections. We used the same 20-minute signal as in §3.1 to create the codebook by the LBG algorithm. The codebook size was 4096. Fig. 5 shows the search accuracy when 10 votes were cast for each original feature vector. It is shown that Probabilistic Dither-voting achieves higher search accuracy than non-dither voting; for example, accuracy improved from 79.5% to 84.0% when the window size is 2.0 second.

Next, we performed a search accuracy test using the query signal (A-1) and the stored signals (A-2,3,4,5,6), where the window size was 5.0 seconds only and $\sigma_Q^2$ is 0 in (A-5,6). Table 1 shows the search accuracy. The Probabilistic Dither-voting achieves higher search accuracy than non-dither voting.

[1](5) is captured in the condition of 64 × 48 image size and 330 kbps JPEG compression. (6) was captured in the condition of 64 × 48 image size and 100 kbps MPEG compression.

## 4. CONCLUSION

This paper has proposed a search method for quickly searching through long video recordings. We used LBG-based VQ to improve the capacity to represent features and Probabilistic Dither-voting to improve robustness with respect to feature distortions caused by video dubbing or low-bit-rate compression. We tested this method in video search experiments using a 1-hour video recording containing several kinds of distortions. The experiment showed that the method improves the search accuracy. Future work will include a further investigation of feature normalization, and an extension to an audio search and a search for signal captured by mobile terminals in the real world.

## 5. ACKNOWLEDGMENTS

The authors thank Drs. Ken'ichiro Ishii, Noboru Sugamura and Norihiro Hagita for their help and encouragement.

## 6. REFERENCES

[1] K. Kashino, G. Smith and H. Murase, ""Time-series active search for quick retrieval of audio and video"," *Proc. of ICASSP-99*, vol. 6, pp. 2993–2996, Mar. 1999.

[2] E. Wold, T. Blum, D. Keislar, and J. Wheaton, ""Content-based classification, search, and retrieval of audio"," *IEEE Multimedia*, vol. 3, no. 3, pp. 27–36, 1996.

[3] Howard D. Wactlar, Michael G. Christel and Alexander G. Hauptmann, ""Lessons Learned from Building a Terabyte Digital Video Library"," *Computer*, pp. 66–72, Feb. 1999.

[4] V. V. Vinod and H. Murase, ""Focused color intersection with efficient searching for object extraction"," *Pattern Recognition*, vol. 30, no. 10, pp. 1787–1797, 1997.

[5] K. Kashino, T. Kurozumi and H. Murase, ""Quick AND/OR Search for Multimedia Signals Based on Histogram Features"," *IEICE Trans. (D-II)*, vol. J83-D-II, no. 12, 2000.

[6] Y. Linde, A. Buzo and R. M. Gray, ""An algorithm for vector quantizer design"," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, 1980.