SCENE RETRIEVAL WITH SIGN SEQUENCE MATCHING
BASED ON VIDEO AND AUDIO FEATURES

Noboru Babaguchi  Tetsuya Ishida  Keisuke Morisawa
2-1 Yamadaoka Suita, 565-0871 Japan
Graduate School of Engineering, Osaka University
babaguchi@comm.eng.osaka-u.ac.jp

ABSTRACT
This paper presents a method of retrieving similar scenes from video streams with sign sequence matching. The visual and auditory streams are partitioned into fixed-length video/audio packets. Based on feature vectors extracted from these packets, sign sequences are formed. The sign sequence can be viewed as abstraction of video and audio features. DP Matching between the target and query sign sequences allows us to find scenes similar to the query in the video stream. For the purpose of efficient processing, packets, histogram based features, and sign sequences, which are the key ideas in this method, are introduced. The preliminary experimental results show that this method is promising for quick retrieval for similar scenes.

1. INTRODUCTION
Video data is composed of temporally synchronized multimodal information streams such as visual, auditory, text and graphics streams. Because these streams are closely related to each other, collaborative analysis focusing on the multimodality could be more useful in extracting video semantics than a single modal analysis[1]. In particular, the auditory stream and the text stream such as transcript data or video caption have rich information about video semantics. We called this kind of strategy of collaborative analysis inter-modal collaboration[2, 3, 4]. The so-called semantic gap prevents us from extracting its semantics from the video data. We think that intermodal collaboration may be one of the most promising strategies to be capable of bridging the gap between the signal level and the semantic level.

Let us here survey approaches that concentrate on multimodality of video streams. We divide approaches into three classes, considering what kind of stream is used. The first class is using the auditory and visual streams. Chang et al.[5] used the word spotting technique to limit the search space of the visual stream. Rui et al.[6] developed a method of extracting baseball highlights through classification of auditory sources.

The second class is using the text and visual streams. For sports video, Babaguchi et al.[2] and Nitta et al.[3] tried to extract events, actions and players from American football broadcasts, using the closed caption and visual streams. Satoh et al. [7] have developed a system for associating faces extracted from the visual stream with their names from the streams of closed caption text and video captions.

The third class is using the auditory, text and visual streams. Smith et al. [8] proposed a method for video summarization based on features extracted from such streams. Huang et al. [9] and Jasinschi et al. [10] proposed methods for segmenting a news video and semantically classifying each segment combining features of all the streams. Li et al. [11] proposed a method of detecting semantically related scenes based on the similarity between the visual and auditory streams, as well as acquiring the semantic contents from the text stream.

In this paper, we propose an efficient method for similar scene retrieval for large-scale video data through collaborative analysis of visual and auditory streams. Large-scale video data means a day/ week/ month-long stream. Since it has a large amount of data, efficiency of processing should be strongly required. We therefore deal with video segments of fixed length instead of shots. For this segment, its temporal slice image is considered for reduction of the computation cost. Subsequently, global video features are extracted from the slice image. On the other hand, audio segments of fixed length are processed independently and audio features are extracted. Both features are abstracted and transformed into sign sequences in a unified way. Retrieval of similar scenes is realized with matching between the target and query sign sequences. The key ideas such as fixed-length video/audio segments, global features, and sign sequences are introduced for the purpose of efficient processing for large-scale video data.

The rest of this paper is organized as follows. Sec.2 outlines the proposed method. Sec.3 and Sec.4 present video and audio features used in the method. Sec.5 presents sign features.
sequence matching. In Sec. 6, we show experimental results. Sec. 7 gives concluding remarks.

2. OUTLINE OF THE METHOD

Fig. 1 shows the outline of the proposed method. We first partition the visual stream into fixed-length segments called video packets [13]. The video packet can be viewed as the partial stream on which a time window of fixed length is operated. Recall that most of the conventional methods are based on shots. The shot is defined as consecutive image frames at a single camera view. Unfortunately, it is impossible to fully segment the visual stream into the shots through image processing. For simplicity of processing, we deal with the video packets. Note that the video packet is fixed length, while the shot is variable. The auditory stream is partitioned into audio packets as well.

For each packet, we extract a variety of video and audio features, forming feature vectors. Clustering is performed on the vector space, and a set of signs, respectively. The process of this method is described as two kinds of transformations: vectors and a set of signs, respectively. The process of this method is described as two kinds of transformations: vectors and a set of signs, respectively. The process of this method is described as two kinds of transformations: vectors and a set of signs, respectively. The process of this method is described as two kinds of transformations: vectors and a set of signs, respectively. The process of this method is described as two kinds of transformations: vectors and a set of signs, respectively.

Fig. 1. Outline of the proposed method.

\( \psi_1 : \mathcal{P} \rightarrow \mathbf{V} \)

In addition, \( \psi_2 : \mathbf{V} \rightarrow \mathcal{A} \).

This indicates that the packet is transformed into its corresponding sign via its feature vector.

Scene retrieval is achieved by retrieval by example. A query sequence is transformed into its sign sequence. Matching between the target and query sign sequences enables us to find a scene similar to the query. The proposed method is characterized as follows: video signs and audio signs are evenly evaluated, and simple but effective features are used for applicability to large-scale video data.

3. VIDEO FEATURES

A video packet can be looked upon as a cuboid whose base is its initial or last image frame and whose height is its temporal length. A pixel in the video packet is denoted by \( F(x, y, t) \), where \( x \) and \( y \) are the spatial axes on the image frame, and \( t \) is the temporal axis. We think that histogram based features are suitable for the video packet because they indicate global and rough features. The following two features are taken into consideration.

a) Tensor histogram: This feature was originally proposed by Ngo et al. [14], reflecting on a rough motion feature in video packets. For a given video packet, we produce two temporal slice images: a horizontal slice \( S^h(u, t) \) and a vertical slice \( S^v(u, t) \), where \( u \in \{x, y\} \). The horizontal and vertical slices represent the horizontal and vertical cross section planes passing through the center of the image frame, respectively.

Let \( S \) be either \( S^v \) or \( S^h \), and \( w \) be a small block of size \( 3 \times 3 \) in \( S \). The structural tensor \( \Gamma \) for \( w \) is defined as

\[
\Gamma = \left[ \frac{\sum_w S_u^2}{\sum_w S_u S_t} \quad \frac{\sum_w S_v S_u}{\sum_w S_v^2} \right]
\]

where \( S_u \) and \( S_t \) are the difference images along \( u \) and \( t \) dimensions, respectively. The local orientation \( \theta \) for \( w \) can be derived from the structural tensor. For details of this derivation, consult [14]. In this case, \( \theta \) ranges between \(-\pi/4\) and \( \pi/4 \). We generate eight bins of the tensor histogram by quantizing the range uniformly, and increment the bin if the pixel in the block has the orientation that the bin specifies. Finally, we obtain an eight-dimensional feature vector from the slice image. Since we here consider the horizontal and vertical slices, two kinds of vectors are generated.

b) Color histogram: This is known as a useful feature in image retrieval because it reflects on the global distribution of color information. To reduce the computational cost, we extract the image frame at some interval from the video packet. For such image frames, after transforming the RGB color space into the HSB space, histogram in terms of the HSB space is considered. We count the number of pixels whose value should fall into the five bins for the three kinds: H, S and B. Totally, a 15-dimensional vector is formed.

As a result, three kinds of feature vectors are formed for each video packet.
4. AUDIO FEATURES

An audio signal that constitutes the auditory stream is also partitioned into audio packets of fixed length. They can be thought of as the partial signal given by a time window of fixed length. For speech recognition, a time window of msec. width is usually used. To acquire semantic information in the audio signal, however, it is pointed out that the time window which is at least 1 sec. should be required [12]. Thus, we assume the packet length is 1 sec.

The signal in the audio packet is expressed as audio frames resulting from sampling. Here lists the audio features for the audio packet.

a) Average short time energy, ASTE: Let \( L \) denote the total number of all audio frames. The short time energy \( STE(n) \) is defined as

\[
STE(n) = \sqrt{\frac{1}{L} \sum_{m} [x(m)w(m-n)]^2}
\]

where \( x(m) \) is a discrete audio signal and \( w(m) \) takes 1 if \( m \) is included in the time window; 0 otherwise. Averaging \( STE(n) \) for \( n = 0, \ldots, L - 1 \), we can obtain \( ASTE \).

b) Voice frame ratio, VFR: All audio frames in the audio packet are classified into voice or background frames. The voice frame is determined as the frame where both its packet are classified into voice or background frames. The short time energy and its amplitude of the spectrum at low frequency are large. \( VFR \) is the ratio of the voice frames to all the frames.

c) Average zero crossing rate, AZCR: The zero crossing rate \( ZCR(n) \) is defined as

\[
ZCR(n) = \frac{1}{2} \sum_{m} |sgn[x(m)] - sgn[x(m-1)]|w(n-m)
\]

where \( sgn[x(m)] = 1 \) if \( x(m) \geq 0 \); \( sgn[x(m)] = -1 \) otherwise. \( AZCR \) is an average of \( ZCR \).

d) Low STE ratio, LSTER:

\[
LSTER = \frac{1}{2N} \sum_{n=0}^{N-1} [sgn[ASTE - STE(n)] + 1]
\]

where \( N \) denotes the number of background frames.

e) Average spectrum envelop power, ASEP: Let SEP denote a power value obtained from a spectral envelop from 4kHz to 11 kHz. \( ASEP \) is an average of \( SEP \) for background frames.

f) Spectrum flux, SF:

\[
SF = \frac{1}{(N-1)(K-1)} \sum_{n=1}^{N-1} \sum_{k=1}^{K-1} (\log(X(n,k))) - \log(X(n-1,k)))
\]

where \( X(n,k) \), \( k = 1, \ldots, K \) is the \( k \)th spectrum at time \( n \).

For the audio packet, we form a six-dimensional feature vector as \( (ASTE, VFR, AZCR, LSTER, ASEP, SF) \).

5. SIGN SEQUENCE MATCHING

We transform the sequences of video and audio packets into sign sequences. To this end, we introduce clustering process. First, we provide a reference space to evaluate the similarity between the video/audio sequence and a query sequence. A variety of example streams with sufficient length are divided into video/audio packets, simply called packets. As stated previously, the features are extracted from packets and four kinds of feature vectors are formed. The reference space corresponding to each kind of feature vector is spanned with example vectors. Note that four reference spaces are individually constructed.

For all the vectors in the reference space, the K-means clustering algorithm, which produces \( K_{\text{max}} \) clusters, is operated. Each cluster is represented as its centroid. Canonical signs \( A_k, k = 1, \ldots, K_{\text{max}} \) are assigned to the clusters in the order of the distance from the origin of the reference space to the cluster.

Next, we consider generating the sign sequence. The target video stream from which we intend to retrieve similar scenes is represented as a sequence of packets. We extract features from each packet, generating its feature vector. This vector is mapped into the corresponding reference space. The cluster nearest to it is found in the nearest neighbor fashion. In this case, the Euclidean distance between the centroid of the cluster and that of the vector is considered. The canonical sign of the nearest cluster is assigned to the packet. In this way, the target sign sequence is generated. On the other hand, the query video sequence is also transformed into the query sign sequence in the same way. The sign sequence may be viewed as abstraction of the video/audio features. We expect that the use of the sign sequences contributes to reducing the computation cost.

Matching between the target and query sign sequences is achieved by the DP matching, which allows us to deal with the sequences in different length. First, we produce a score matrix whose elements depend on the distance among the centroids of all clusters. Using this matrix, we determine a temporal position in the sign sequence where the edit distance is minimized. The edit distance is defined as the sum of the scores required for unification of the two sign sequences by repeating three operations: insert, delete and exchange. This matching algorithm is frequently exploited in DNA sequence alignment. Finally, we can get the temporal position where the target scene begins.

6. EXPERIMENTAL RESULTS

The experiment was conducted for actual broadcast video whose genre was the Japanese entertainment program. The parameters were set as follows. The length of a video packet was 20 sec., while that of an audio packet was 1 sec. The
image frame is of size 352 x 240. The temporal slice was of size 352 x 600 (horizontal), and 240 x 600 (vertical). The image frame rate was 30 frames per second. On the other hand, audio signals were sampled at 48 Kbps, quantized by 16 bits. The audio frame length \( L \) was 512. At the clustering stage, the number of clusters was four, i.e. \( K_{\max} = 4 \). In other words, four kinds of sign symbols were used.

The target was three video streams of a daily TV program, which were broadcasted in different days. The length of each video was about four hours. We tried to retrieve similar scenes when we gave a scene corresponding to the time interval of a meaningful unit in the same program in the other day, as a query. The length of the query was about 15 min. In reality, each video had a single scene that should be retrieved.

We examined the following three cases: i) retrieval with video features, ii) retrieval with audio features, and iii) retrieval with video and audio features. For each case, we tried 20 retrievals with seven different queries. Retrieval results were ordered according to their matching score. Correct detection is assumed if some results of top-five retrievals were ordered according to their matching score. Correlation position of the scene. The evaluation metrics are the recall rate \( R_C \) and the precision rate \( P_C \). Also, we consider the F-value \( F = 2 \cdot R_C \cdot P_C / (R_C + P_C) \).

Table 1 summarizes the retrieval results for each case. We were not able to obtain good results with video features only. Because four sign symbols were taken into account in this case, similar sign sequences sometimes appeared. We guess that the number of sign symbols that depends on the cluster number is too small. The appropriate number of clusters should be examined for video contents. In contrast, satisfactory results were obtained for scenes including music because the audio features used in this method were suitable for identifying music. In using both video and audio features, we identified both results within \( T\text{-diff} \). Table 1 indicates that for \( T\text{-diff} = 180 \) sec., the F-values for video, audio, and video & audio are 15, 30, and 52, respectively. Since the collaborative retrieval can take the highest F-value, retrieval based on multimodal cues turns out very effective. More experiments for large-scale video will be our future work. In addition, the packet length, the number of clusters and the DP parameters should be explored.

### Table 1. Retrieval results.

<table>
<thead>
<tr>
<th>T-Diff</th>
<th>60sec</th>
<th>120sec</th>
<th>180sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>i) Video</td>
<td>( R_C )</td>
<td>10%</td>
<td>35%</td>
</tr>
<tr>
<td></td>
<td>( P_C )</td>
<td>2%</td>
<td>7%</td>
</tr>
<tr>
<td>ii) Audio</td>
<td>( R_C )</td>
<td>55%</td>
<td>70%</td>
</tr>
<tr>
<td></td>
<td>( P_C )</td>
<td>11%</td>
<td>14%</td>
</tr>
<tr>
<td>iii) Video&amp; Audio</td>
<td>( R_C )</td>
<td>30%</td>
<td>45%</td>
</tr>
<tr>
<td></td>
<td>( P_C )</td>
<td>18%</td>
<td>26%</td>
</tr>
</tbody>
</table>

### 7. CONCLUSIONS

We have addressed the method of retrieving similar scenes with sign sequence matching. The preliminary experimental results show the favorable property of this method. The matching of the sign sequences resulting from feature extraction from for visual and auditory streams enables efficient retrieval reflecting on video semantics. For the purpose of efficient processing, the key ideas such as packets, histogram based features, and sign sequences, are embodied in this method. The remaining work is to apply this method to much longer streams and to check its scalability.

### 8. REFERENCES


