Key-frame Extraction of Wildlife Video based on Semantic Context Modeling

Suet-Peng Yong
Department of Information Science
University of Otago, New Zealand;
Universiti Teknologi PETRONAS,
Perak, Malaysia.
yongsuetpeng@petronas.com.my

Jeremiah D. Deng
Department of Information Science
University of Otago
Dunedin, New Zealand
jeremiah.deng@otago.ac.nz

Martin K. Purvis
Department of Information Science
University of Otago
Dunedin, New Zealand
mpurvis@infoscience.otago.ac.nz

Abstract—In recent work on image and video retrieval there seems to be a shift of focus from low-level feature extraction to producing high-level semantic representation of scenes. This paper presents a framework that produces semantic context features from video frames which are then employed for key-frame extraction. Working with wildlife video frames, the framework starts with image segmentation, followed by low-level feature extraction and classification of the image blocks extracted from image segments. Based on the image block labels in the neighbourhood a co-occurrence matrix is then constructed to represent the semantic context of the scene. The semantic co-occurrence matrices then undergo binarization and principal component analysis for dimension reduction, forming the feature vectors used in a one-class classifier that extracts the key-frames. Experiments show that the utilization of high-level semantic features result in better key-frame extraction when compared with methods using low-level features only.

Index Terms—image semantic analysis, classification, video summarization

I. INTRODUCTION

One of the important procedures involved in video summarization and content-based video retrieval is key-frame extraction. Key-frame extraction provides a suitable visual summary by selecting a set of key-frames from video sequences. Key-frames can also represent the salient content and shots in video. Automatic key frame extraction can also facilitate a number of multimedia applications such as content management, editing, and surveillance, reducing the need of traversing the long video clips.

In the past, a variety of low-level features used in image analysis, such as colour histograms [1], [2], Gabor texture features [3], [4] and template shape structures [5] have been used for key-frame extractions. However, the semantics of the video content cannot be expressed directly by these low-level features. It has been an ongoing research to bridge the semantic gap from low-level features to semantic content descriptions that are necessary to leverage multimedia data analysis and management.

On the other hand, recent psychophysical studies have revealed the importance of high-level semantics in object recognition and scene interpretation. It has been shown that top-down facilitation in recognition is triggered by early information about an object, and also by contextual associations between the objects [6], [7]. Based on the object contextual information, high-level feature has shown promising results for scene understanding [8]. Such kind of semantic features can also be extended for representing content of video frames.

In this paper, we consider the problem of key-frame extraction in wildlife video. We propose a computational framework that models the semantic context in video sequences, where key-frame extraction can be conducted based on monitoring the change of semantic features between video frames. By employing a statistical one-class classifier on the semantic feature distances, key-frames can be extracted as outliers.

In the remainder of the paper, we first introduce our proposed key-frame extraction model in Section II. An empirical study comparing the proposed method with some conventional methods is then presented in Section III. We briefly review some related work and highlight our contribution in IV. Finally we give the conclusion and outline a few future directions.

II. COMPUTATIONAL FRAMEWORK FOR KEY-FRAME EXTRACTION

The proposed computational framework for key-frame extraction is shown in Figure 1. We start with image segmentation, followed by low-level feature extraction and classification of the image blocks extracted from the segmented images. The
labeled image blocks then generate a co-occurrence matrix of object labels, representing the semantic context of the scene. After going through further processing such as binarization or principal component analysis (PCA), the co-occurrence code then serves as a high-level semantic feature to be used for frame analysis. Finally we employ a one-class classifier for key-frame extraction, taking outliers as potential key-frames.

A. Image Segmentation

A scene usually contains multiple objects of different visual characteristics. By segmenting an image into homogeneous regions, it facilitates detection or classification of these objects. JSEG algorithm [9] was used for this purpose. In JSEG, colors in the image are first quantized to several representing classes that are used to separate the regions in the image. Image pixel colors are then replaced by their corresponding colour class labels to form a class-map of the image. In order to get good segmentation, the high and low values correspond to possible boundaries and interiors of colour-texture regions are adjusted until the object and background can be differentiated. Next, those segments that exceed a threshold will be taken out and stored as individual segment images respectively. Those segments will then be annotated manually as the ground truth for training classifiers. Each segment image will be tiled into blocks of size 20×20. The image blocks that fall out of the segment edges will be ignored. Due to visual heterogeneity within segments, we have found it is more effective to train classifiers on segment blocks instead of segment images directly.

B. Feature Extraction

After extracted image blocks from the segmented image, visual features are then generated for the image blocks. First we employ the LUV colour histogram to encode the colour information of image blocks. Colour histograms are found to be robust to resolution and rotation changes, and we adopt the LUV colour space as it models human’s perception of colour similarity very well [10]. We quantize the LUV channels using the same interval, thus the L channel has 20-bins, U has 70-bins and V has 52-bins respectively. The standard deviation of the LUV histogram values is also calculated and included. The LUV histogram feature therefore has 143 dimensions.

Texture features extracted from the image blocks are also included as local features. We adopt the Haralick features which consist of a few statistical properties based on the gray scale co-occurrence matrix [11]. The image block is first converted to gray scale. The co-occurrence matrix is a two-dimensional histogram composed from the pairwise statistics of gray scale co-occurrence among adjacent pixels. Four orientations are considered, each giving a co-occurrence matrix for an image block. A total of 13 statistical measures can be calculated for each co-occurrence matrix. The mean and deviation values of each of these 13 measures over the four orientations give a 26-dimension feature code. Finally, the colour and texture features are concatenated together, giving a feature vector of 169 dimensions to represent an image block. Through manual labeling of image segments for the training images, semantic ground truth is assigned to image segments. The image blocks inherit semantic labels from their corresponding segments. These feature codes along with the relevant class labels are used to train the object classifiers.
C. Semantic Context

To model the semantic context within a scene, we further generate a block label co-occurrence matrix (BLCM). First the labels of all image blocks should be obtained - either by manual annotation for training images, or automatic classification when testing. Then the image blocks are scanned from left to right and top to bottom, the co-occurrence of labels for blocks within a distance threshold \( R \) is collected. We set \( R = 150 \) in our experiment. The co-occurrence statistics is gathered across the entire image frame and normalized by the total number of image blocks. The variation on the object sizes affect the matrix values of BLMC, so to reduce this effect the BLMC is binarized. Figure 2 shows a ‘lionness’ image as an example, with blocks labels and the relevant binary BLMC shown below. There are 4 object labels in the training data: ‘grass’, ‘lion’, ‘sky’ and ‘zebra’, hence giving a \( 4 \times 4 \) BLMC.

The resulting BLMC is consequently asymmetric. This is however a desirable feature as we intend to keep the relative spatial location information in the semantic context representation. For example, a scene context in Figure 3 may have the entry of ‘lion’\( \rightarrow \)‘zebra’ (scanning top-down), while a ‘zebra’\( \rightarrow \)‘lion’ BLMC entry in Figure 4 may suggest some different semantics. We concatenate the BLMC matrix rows into a 1-D vector, and PCA can then be used for dimension reduction. The BLMC code with or without PCA then serve as the semantic feature for frame/scene representation to be used later.

D. Key-frame Extraction

Denote the \( i \)-th frame of video sequence as \( I_i \). The BLMC feature representing frame \( I_i \) is extracted as \( f_i \). The Manhattan distance is calculated between the features of the current frame and its preceding frame, as follows:

\[
d_i = || f_i - f_{i-1} ||. \quad (1)
\]

We call this distance ‘Adjacent Frame Distance’ (AFD). The mean distance \( \mu \), and standard deviation \( \sigma \) of AFD across the entire video sequence are then obtained.

For most video frames within a shot, the distance between adjacent frames will be small as the BLMC code do not change much. However, when significant changes do happen for a new frame, it must present some new semantic content and hence can be picked as a key frame. Hence, we can use a one-class classifier with a threshold \( T \) to classify a frame into the key-frame set \( \mathcal{F} \), as follows:

\[
\mathcal{F} \leftarrow \mathcal{F} \cup I_i, \text{if } d_i > T, \quad (2)
\]

where the threshold can be set as

\[
T = \mu + k\sigma, \quad (3)
\]

where \( k \in \mathbb{R} \). The value of \( k \) controls the number of potential key-frames of a video sequence. Assuming AFDs are of a Gaussian distribution, about 95\% of the values lie within two standard deviations from the mean, hence with \( k = 2 \) there is a probability of about 2.3\% for the distance to be greater than the threshold \( T \), and consequently the relevant frame will be chosen as a key-frame.

III. EXPERIMENTS AND RESULTS

Experiments were conducted on eight wildlife video clips downloaded from YouTube. These videos mostly have the storyline of animals confronting and/or hunting their preys. Our key-frame extraction approach aims at capturing those inter-changing positions of the animals in the videos. The number of frames for each video and its resolution are listed in Table I.

In order to recognize the objects in those video frames, images taken from the frames of other videos in YouTube with similar resolution and scene types were taken as training images. The objects domain in those videos consist of ‘grass’, ‘sky’, ‘water’, ‘lion’, ‘zebra’, ‘buffalo’, ‘tiger’, ‘cow’ and ‘deer’.

Key-frames are extracted based on the method given in Section II-D. For comparison, we also apply the same one-class classification method to the AFD values derived from the low-level features of sequential video frames. The 169-dimension low-level feature vector is used, incorporating a global LUV colour histogram and the Haralick texture features. Apart from that, our method was also compared with the Gaussian Mixture Model (GMM) clustering method [12].

For fair comparison, the number of key-frames acquired by different methods are kept roughly the same.

A. Computational results

The graphs of AFD based on the low-level features and the BLMC code for ‘lionzeb1’ and ‘lionzeb2’ are shown in Figure 5 and Figure 6 respectively.

In order to reduce frame redundancy and to keep the number of key-frames extracted to be less then 2\% of the total number of frames, a key-frame interval threshold is applied across the video. This interval threshold is set as 65. A \( k = 0.5 \) setting is applied to ‘lionzeb1’ whereas for ‘lionzeb2’, \( k = 2 \). A higher \( k \) is used for ‘lionzeb2’ due to the fact that ‘lionzeb2’ has many more frames than ‘lionzeb1’ and we want to keep the number of key-frames down. As can be seen from the graphs in Figure 5(a) and Figure 6(a), with low-level features, not many frames above the threshold are found and classified as key-frames, hence quite a few interesting frames have been missed out.
Key-frame sets were generated by the following methods: AFD using BLCM code, AFD using low-level features and GMM. The respective key-frames extracted from each approach for ‘lionzeb1’, ‘lionzeb2’ and ‘lionbuf’ are shown in Figure 7, Figure 8 and Figure 9 respectively.

B. Subject tests

We also conducted subject tests to evaluate and compare the effectiveness of different key-frame extraction methods. Subject tests have been suggested as the most useful and realistic form of evaluation in video abstraction [13]. To evaluate the quality of the extracted key-frame sets, we followed the approach similar to that of [14]. This was conducted individually on twelve to eighteen human subjects for each video respectively. The subjects are all university students and the test was carried out double-blindly. Prior to the test, every subject was informed that key-frames mean the video frames that can represent the event occurred and provide informative representation of the video. In other words, by looking at the key-frames, one can grasp the gist of a video clip without viewing the full video.

In the test, a subject was first shown a testing video clip. After that, the three sets of key-frames extracted from the particular video clip were shown to the subject. The subject was then requested to rank his or her preference of the key-frame sets that would best summarize the video clip. The three key-frame sets were ranked from 1 to 3 with 1 being the best and 3 the worst. The average rank scores for each method applied to each video clip were then calculated. This process was carried out for eight video clips and the results are as shown in Table II. The average ranks of different methods are also summarized in Table II. BLCM has the smallest average rank at 1.83, followed by GMM at 2.02, while the average rank for colour-texture sits the highest but not much different from GMM, that is 2.14.

Kappa is a statistical index that measures the inter-rater reliability for categorical items [15]. It is the preferred statistic that accounts for chance on how likely two or more observers agree in their interpretations. The Kappa agreement values are listed in Table III. In five out of the eight video clips, the Kappa values show moderate inter-rater agreement, while others show
To further verify whether the ranking of BLCM-generated key-frames are significantly better, the Mann-Whitney U-test was conducted pair-wise on results obtained by BLCM and the other two methods. The results are reported in Table IV. The test rejects the null hypothesis of equal medians at the default 5% significance level for BLCM vs colour-texture, but not for BLCM vs GMM. It seems that BLCM is statistically better ranked as compared with low-level features; however, its rank is similar to the GMM approach.

It is noticed that the key-frames retrieved by GMM and the low-level features miss out the interesting frame of the lioness attacking the zebra in ‘lionzeb1’, but it can be captured by BLCM code, e.g., with Frame 136 in Figure 7(b).

For the longer clip ‘lionzeb2’, some of the frames extracted by using the low-level feature and GMM are obviously transitional frames, such as Frames 1189 and 1601 in Figure 8(a); and Frames 1534 and 3487 in Figure 8(c), all bearing little semantic content. In contrast, it can be seen that the interesting frames showing the important actions context about the lioness...
Figure 8. Key frames extracted from 'lionzeh2' using (a) low-level features, (b) AFD on BLCM, and (c) GMM.
Figure 9. Key frames extracted from ‘lionbuf1’ using (a) low-level features, (b) AFD on BLCM, and (c) GMM.

confronting the zebra are captured by the BLCM code, e.g. Frame 2848, 2920 and 3012 in Figure 8(b). The proposed method of AFD using BLCM code also works the best for ‘lionbuf1’, picking out more fighting scenes that represent the dynamics of the video.

To summarize, as seen from the experiment outputs, the key-frames extracted by the proposed framework can better locate the semantic changes between the video scenes. On the other hand, key-frames extracted by the low-level feature counterpart may extract more key-frames of varying visual appearance, which probably explains why they may be favoured by some subjects as shown in the test for ‘lionzeb2’. To some users, the inclusion of such frames seems to improve the summarization, while on the other hand, other subjects may not prefer those frames due to their lack of semantic interestingness.

IV. RELATED WORK

Conventionally, key-frame extraction techniques involved shot segmentation [17], [18]. The shot transition is usually indicated when the difference of the adjacent frames features exceeds a certain threshold. Nevertheless, some shots may be meaningless even though the transition frames between shots show significant differences. Although the local visual semantics from region thesaurus [18] can model the semantic content of video frames and therefore used for key-frame extraction, its representation cannot always efficiently capture the content of a shot, and the clustering process to group a set of frames has also a major issue in computational complexity.

In an algorithm proposed for sequential key-frame extraction [19], texture feature of DCT coefficients and colour layout feature for consecutive frames are compared in sequential manner, and key-frames are detected depending on the dissimilarity with the previously detected key-frames based on a threshold. The drawback of the method is the lack of guideline
in selecting the threshold also the lack of control on the number of extracted key-frames. Our work differs from it by using semantic features.

In terms of the application domains of video summarization, most of the work focuses on the domain of sports video. In order to model the semantics of video frames, object based and event based models in video summarization have been proposed for sports events [20]. For example, object features are used to detect referees and the penalty box in soccer games [21], and players detection in fighting sports video was conducted to determine key-frames based on the distance between players [22], and a video understanding scheme for baseball programs can be established similarly [23].

Animal hunt detection was proposed by Haering et al. [24] in wildlife video. Their event-based method utilized a combination of colour, texture and motion features and employs neural network classifiers with an event inference model, thus raising the issue of computational complexity. Another visual abstraction work by Gibson et al. [12] on wildlife footage using Gaussian Mixture Models was able to produce promising video abstraction, however, it did not handle semantic analysis in its computational model.

In contrast, our model can detect interesting frames based on the objects information and context modeling without the requirements of motion features or pre-trained events. On top of that, a one-class classifier proposed in our model that utilizes the statistical approach is able to align with the intended amount of interesting key-frames.

V. CONCLUSION

In order to close up the semantic gap in video analysis, it is essential to derive semantic features that represent the high-level content of the video frames. Our proposed approach employs a one-class classifier operating on semantic context features, and our preliminary results on key-frame extraction have demonstrated that interesting frames can be more effectively extracted using semantic cues, achieving better performance compared with using low-level features alone.

On the other hand, the success of our proposed framework does rely on the accuracy of image segmentation and image block classification. Further exploration of better approaches in object segmentation and recognition from video frames are necessary to strengthen the model. Also we have focused on object detection and context modeling in this work, but motion features can be integrated in the framework so as to include events into the semantic representations. Upon further improvement of our proposed framework, we believe it will offer a promising solution to semantic video summarization and retrieval.

REFERENCES


[22] N. Benjamas, N. Cooharojananone, and C. Jaruskulchai, “Flashlight and event based models in video summarization have been proposed for sports events [20]. For example, object features are used to detect referees and the penalty box in soccer games [21], and players detection in fighting sports video was conducted to determine key-frames based on the distance between players [22], and a video understanding scheme for baseball programs can be established similarly [23].

Animal hunt detection was proposed by Haering et al. [24] in wildlife video. Their event-based method utilized a combination of colour, texture and motion features and employs neural network classifiers with an event inference model, thus raising the issue of computational complexity. Another visual abstraction work by Gibson et al. [12] on wildlife footage using Gaussian Mixture Models was able to produce promising video abstraction, however, it did not handle semantic analysis in its computational model.

In contrast, our model can detect interesting frames based on the objects information and context modeling without the requirements of motion features or pre-trained events. On top of that, a one-class classifier proposed in our model that utilizes the statistical approach is able to align with the intended amount of interesting key-frames.

V. CONCLUSION

In order to close up the semantic gap in video analysis, it is essential to derive semantic features that represent the high-level content of the video frames. Our proposed approach employs a one-class classifier operating on semantic context features, and our preliminary results on key-frame extraction have demonstrated that interesting frames can be more effectively extracted using semantic cues, achieving better performance compared with using low-level features alone.

On the other hand, the success of our proposed framework does rely on the accuracy of image segmentation and image block classification. Further exploration of better approaches in object segmentation and recognition from video frames are necessary to strengthen the model. Also we have focused on object detection and context modeling in this work, but motion features can be integrated in the framework so as to include events into the semantic representations. Upon further improvement of our proposed framework, we believe it will offer a promising solution to semantic video summarization and retrieval.

REFERENCES
