

Video Summarization with Supervised Learning

Jayanta Basak
IBM India Research Lab
New Delhi, India
bjayanta@in.ibm.com

Varun Luthra and Santanu Chaudhury
Dept Electrical Engineering
Indian Institute of Technology, New Delhi, India
Santanuc@ee.iitd.ac.in

Abstract

We present a video summarization technique based on supervised learning. Within a class of videos of similar nature, user provides the desired summaries for a subset of videos. Based on this supervised information, the summaries for other videos in the same class are generated. We derive frame-transitional features and subsequently represent each frame transition as a state. We then formulate a loss functional to quantify the discrepancy between state transitional probabilities in the original video and that in the intended summary video, and optimize this functional. We experimentally validate the performance of the technique using cross-validation scores on two different class of videos, and demonstrate that the proposed technique is able to produce high quality summarization capturing the user perception.

1 Introduction

Video summarization can be broadly classified into two categories namely, static summarization and video skimming. In this paper, we address a static video summarization scheme with supervised learning, and do not consider the additional audio or text information that may be available in the video. A few representatives of the various techniques for static video summarization available in the literature include time-constrained clustering and hierarchical scene transition graph (HSTG), tree-structured video-table-of-contents (V-TOC) [8], polygonal high-dimensional curve representation of the video trajectory and curve splitting [1], graphical representation of video flow and subsequent processing [7], video flow characterization [2], use of self-organization map [9]. In general, some gap may exist between the automated characterization of the important video frames and the human perception about the importance of a frame. The user feedback has also

been taken into account in generating video summaries such as capturing user interest through user-log activity with a hidden Markov model [10], shot-rank measure [11], and reinforcement learning [6].

In our summarization scheme, a user provides the desired summaries ('ideal' summary) for a subset of video data within a class of video datasets of similar nature (for example, party video). Based on this supervised information, the summaries of the rest of the video datasets are generated. We first obtain the frame-transitional features, and then represent each frame transition as a state of the video stream. We then model the transitional characteristics in the original video and that in the summaries. We then define an objective functional using these state transitional characteristics. We obtain summaries of the test videos by maximizing the objective. Experimentally, we show the effectiveness of the summarization scheme through statistical cross-validation on party videos and soccer videos. The proposed technique not only captures the user perception through the process of supervised learning, but also provides an algorithmic framework for generation of personalized video summary.

2 Video Features

We extract the frame-transitional features in terms of hue similarity, edge similarity, texture similarity, and optic flow between every pair of consecutive frames.

Hue Similarity: We consider the hue similarity between two frames as

$$similarity(h, g) = \frac{\sum_{i=1}^n \min(h(i), g(i))}{\max(|h|, |g|)} \quad (1)$$

where h and g are the hue histograms of two frames, $|\cdot|$ being the magnitude sum. We divide the frames into four quadrants and use concatenated histograms.

Edge Similarity: We extract edge vectors using Sobel operator and then use the histogram similarity measure by quantizing the edge directions into five equally

spaced bins. We also divide each frame into four quadrants and use concatenated histograms.

Texture Similarity: We divide each frame into four blocks and measure the similarity of mean intensity and intensity variance. We compute the output of 25 Gabor filters with five orientations $[0, \pi/6, \pi/3, \pi/2, 3\pi/4]$ and five frequencies $[0, 2, 4, 8, 16]$, and for each filter outcome we measure the similarity. Thus we obtain 27 different features for texture similarity.

Optic Flow: We partition each frame into 64 equal non-overlapping blocks, and measure the mean x -velocity map, the mean y -velocity map and the mean velocity direction map between the two frames for each block using the Lukas-Kanade optic flow computation method [5]. We thus obtain $3 \times 64 = 192$ features for the optic flow.

We thus represent each frame transition by a feature vector of length 221. We map the feature vectors onto a one-dimensional self-organizing map (SOM) [4] such that each video is described as a discrete time-series. In this time series, each frame-pair transition is represented by a node in the SOM. Thus if the SOM has C elements then we represent each frame pair transition by an index $i \in \{1, 2, \dots, C\}$. We denote this index i as the state of the frame transition.

3 Learning

Considering the entire video data (both training test sets) within a class, we obtain the state transition probability of the frame transitions indexed with i and j (in SOM) as $P_{ij} = (N_{ij} + \alpha_{ij}) / (\sum_{j=1}^C N_{ij} + \sum_{j=1}^C \alpha_{ij})$ where N_{ij} is the number of immediate transitions from state i to state j in the discrete time series of all videos. The constant α_{ij} is a Dirichlet prior such that we assign certain non-zero prior to every transition probability. The non-zero prior in the transition probability is useful in the objective functional that we derive for the purpose of supervised learning as discussed later.

We obtain the states of only summary frame transitions in the training set. For example, for a summary $\{s_1, s_2, s_3, \dots, s_m\}$, we get the feature vectors of the frame transitions $\{s_1, s_1+1\}, \{s_2, s_2+1\}, \{s_3, s_3+1\}, \dots, \{s_m, s_m+1\}$ of the original video, and then map these feature vectors to the SOM to observe which states are preserved in the summary. We obtain the state transition probabilities only in the summary frames as $Q_{ij} = (M_{ij} + \alpha_{ij}) / (\sum_{j=1}^C M_{ij} + \sum_{j=1}^C \alpha_{ij})$ where M_{ij} is the number of transitions from state i to state j in the entire set of available video summaries (training set).

We observe that in the set of training videos, if Q_{ij} takes a high value then the state pair $\{i, j\}$ summarizes

a state subsequence $\{i, i_1, i_2, \dots, i_p, j\}$ of an original video. From the human perception point of view, we can expect that there exist some frame i_l or a set of frames in this subsequence which convey some interesting information. From the information gain perspective, we can therefore expect that $P_{i i_1} \prod_{q=1}^{p-1} P_{i_q i_{q+1}} P_{i_p j}$ should be low. Otherwise if all state transitions in the state subsequence are highly expected then the frame pair $\{i, j\}$ may not appear in the summary according to the human perception. We generalize this concept and measure the Kullback-Leibler divergence $D(Q||P)$ over all subsequences that are to be summarized, and maximize the divergence. Formally if $S = \{s_i | i = 1, 2, \dots, m; s_i < s_{i+1}\}$ represents an m -frame summary of a test video then we obtain the objective measure I as

$$I(S) = \sum_{i=1}^{m-1} Q_{c_{s_i} c_{s_{i+1}}} \log \left(\frac{Q_{c_{s_i} c_{s_{i+1}}}}{\prod_{v=s_i}^{s_{i+1}-1} P_{c_v c_{v+1}}} \right) \quad (2)$$

where $c_i \in \{1, \dots, C\}$ represents the state corresponding to the frame transition feature vector $\{i, i+1\}$. For a test video, we obtain the summary S that maximizes $I(S)$. For example, if we have a four-frame summary consisting of frame indices $\{a, b, c, d\}$ then $s_1 = a, s_2 = b, s_3 = c, s_4 = d$, and we compute $I(a, b, c, d)$. The task is to obtain the particular indices $\{a, b, c, d\}$ such that $I(a, b, c, d)$ is maximized.

Maximization of I is a highly complex non-convex optimization task. We apply evolutionary stochastic search specifically genetic algorithm [3] for this purpose. The search space for $S = \{s_1, s_2, s_3, \dots, s_m\}$ is constrained by $s_i < s_j$ for any $i < j$, and $s_m \leq n$ where n is the length of the test video. We transform the constrained search problem into an unconstrained search of integer strings (we represent each integer with their binary equivalents) of length $m+1$, $x = [x_1, x_2, x_3, \dots, x_{m+1}]$. To evaluate each string, first we transform x to y as $y = [y_i | y_i = \sum_{j=1}^i x_j; i = 1, 2, \dots, m, m+1]$, and obtain the candidate summary corresponding to x as

$$s = [s_i | s_i = \lceil \frac{ny_i}{y_{m+1}} \rceil; i = 1, 2, \dots, m] \quad (3)$$

where n is the original length of the video. Since we normalize y_{m+1} to n , the search space constraints are always satisfied. However, search in the unconstrained space in this manner, does not guarantee the distinctiveness of the frames in the summary, i.e., the effective summary length can be reduced to less than m also. In this respect, we mention that the maximum allowable limit of x_i (depending on the number of bits allocated in the string for each integer) plays a significant role in the

granularity of representation of the summary. If a low maximum value is considered then there will be problems of localization of the frames in the summary because a unit increment of x can result in the skip of several frames. On the other hand, a very large maximum value can result in the repetition of the same frames in the summary because of the rounding effect, and the effective length of the summary can decrease. We choose the maximum allowable limit of x to be the same as n , the length of the original video.

4 Experimental Results

We experimented with two different class of video datasets namely, home-shot party video and soccer video, each class containing 50 different video datasets. Home shot party video duration varies from 3 to 15 minutes whereas soccer video duration varies from 2 to 8 minutes. We objectively evaluate the algorithm with k -fold cross-validation for $k = 2, 5$, and 10 respectively. The cross-validation that we do is similar to that used in the pattern classification domain except that we use precision and recall as the performance measures.

$$\text{Precision} = \frac{\text{Number of desired frames in the summary}}{\text{Length of the summary generated}(m)}$$

$$\text{Recall} = \frac{\text{Number of desired frames in the summary}}{\text{Length of the ideal summary}}$$

‘Recall’ indicates the fraction of the number of ‘ideal’ frames in the generated summary, and ‘Precision’ indicates the fraction of the number of generated summary frames that are ‘ideal’. In the k -fold cross-validation, we obtain the precision and recall indices for every trial on every video for different lengths, m , of the intended summary, and we obtain the scatter-plot of the precision-recall measures for all videos within a class. We then show the average characteristics of the precision-recall by fitting a polynomial curve. We used a one-dimensional SOM consisting of 100 elements (neurons) for all videos. In computing the state transition probabilities, we use $\alpha_{ij} = 1$ for all i and j . In the genetic algorithm, we used a population of 150 binary strings (chromosomes), and used a single-point crossover probability 0.8 and a mutation rate 0.05. Note that, the results were similar for a considerable range of parameters values such as length of the SOM and the population size. We used 500 GA iterations (which took 30 seconds approximately). We observed that the results become marginally better after 500 iterations. Figure 1 illustrates two example summaries that are generated.

We show the 10-fold, 5-fold, and 2-fold cross-validation performance in terms of the precision-recall indices for the home video class and the soccer video class in Figures 2 and 3 respectively. In all these cases,



Figure 1. Example summaries generated with 500 iterations of the genetic algorithm illustrating (a) a ten-frame summary, and (b) a twenty-frame summary.

we show the average performance by fitting a high-degree polynomial curve on the scatter-plot. For party videos (Figure 2), we varied m (the intended summary length) from 5 to 100 in steps of 5, and obtained the precision-recall measures for all 50 videos for every value of m . For soccer videos, we varied m from 3 to 60 in steps of 3 (soccer videos are of relatively shorter duration). Since we perform k -fold cross-validation, the performance is not influenced by any subjective judgment after the summarization. Both Figures 2 and 3 reveal that precision is high for a fairly large range of recall. For example, we get almost 85% of the frames in the ideal summary (i.e., recall is 0.85) with a fairly high precision of approximately 0.75 in both the cases, i.e., the intended summary length is only 33% ($1/0.75 = 1.33$) larger than the number of frames retrieved from the ‘ideal’ summary. This indicates that the summarization algorithm is able to capture the user perception to a fairly large extent. Moreover, the similar characteristics of the precision-recall curves in the two completely different categories reveal that the algorithm is applicable to different domains of videos to an extent.

5 Conclusions

We presented a supervised learning based video summarization technique. Within a class of similar video datasets, a user provides the intended summaries

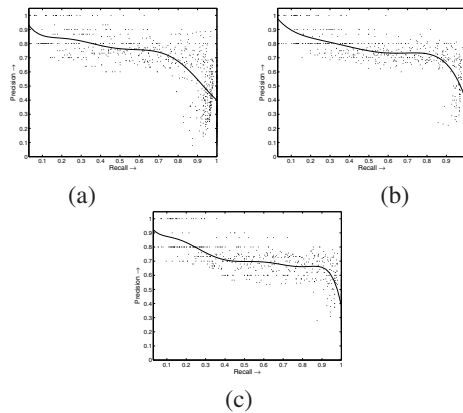


Figure 2. Precision-vs-Recall for the class of home-shot party video dataset for (a) 10-fold cross-validation (CV), (b) 5-fold CV, and (c) 2-fold CV.

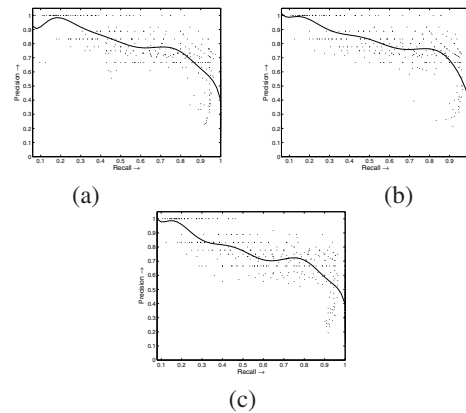


Figure 3. Precision-vs-Recall for the class of soccer video dataset for (a) 10-fold cross-validation (CV), (b) 5-fold CV, and (c) 2-fold CV.

(‘ideal’ summary) of a subset of videos, and based on this supervised information, the summaries of the rest of the videos (test videos) are generated. We derived certain inter-frame transitional features and then characterized each frame-transition by a state. Subsequently we defined an objective functional based on the supervised information using KL-divergence and maximized the objective to obtain summaries for the test videos. We objectively evaluated the performance using k -fold cross-validation, and observed that the algorithm is able to maintain a fairly high precision for a large range of recall for two different class of videos. We derived only image based features; however, audio features can also be incorporated to enhance the performance of the algorithm, the investigation of which constitutes a scope of further study. The nature of the video summaries generated here depends on the desired (‘ideal’) summaries of the training videos. Thus the technique can also be viewed as a means of generating personalized video summaries.

References

- [1] D. DeMenthon, V. Kobla, and D. Doermann. Video summarization by curve simplification. In *Proc. Sixth ACM Intl Conf. Multimedia*, pages 13–16, Bristol, UK, 1998.
- [2] A. M. Ferman and A. M. Tekalp. Efficient filtering and clustering methods for temporal video segmentation and visual summarization. *Journal of Visual Communication and Image Representation*, 9(4):336–351, 1998.
- [3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing, Inc., Reading, Massachusetts, 1989.
- [4] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 1988.
- [5] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence (IJCAI 81)*, pages 674–679, Vancouver, British Columbia, Canada, 1981.
- [6] K. Masumitsu and T. Echigo. Video summarization using reinforcement learning in eigen-space. In *Proc IEEE Intl Conf Image Processing, vol. 2*, pages 267–270, Vancouver, BC, Canada, 2000.
- [7] C. W. Ngo, Y. F. Ma, and H. J. Zhang. Video summarization and scene detection by graph modeling. *IEEE Trans on Circuits and Systems for Video Technology*, 15(2):296–305, 2005.
- [8] Y. Rui, T. S. Huang, and S. Mehrotra. Constructing table of content for videos. *ACM Journal Multimedia Systems (Special Issue Multimedia Systems on Video Libraries)*, 7(5):359–368, 1999.
- [9] A. Stefanidis, P. Partsinevelos, P. Agouris, and P. Doucette. Summarizing video datasets in the spatiotemporal domain. In *Proc Eleventh Intl Workshop Database and Expert Systems Applications (DEXA)*, pages 906–912, Greenwich, London, UK, 2000.
- [10] T. F. Syeda-Mahmood and D. Ponceleon. Learning video browsing behavior and its application in the generation of video previews. In *Proc. Ninth ACM Intl. Conf. Multimedia*, pages 119–128, Ottawa, Ontario, Canada, 2001.
- [11] B. Yu, W. Y. Ma, K. Nahrstedt, and H. J. Zhang. Video summarization based on user log enhanced link analysis. In *Proc. Eleventh ACM Intl. Conf. Multimedia*, pages 382–391, Berkeley, CA, USA, 2003.