I Remember Seeing This Video: Image Driven Search in Video Collections

Zheng Wang
Faculty of Science
University of Ontario Institute of Technology
Oshawa ON L1H 7K4 Canada
zheng.wang@uoit.ca

Faisal Z. Qureshi
Faculty of Science
University of Ontario Institute of Technology
Oshawa ON L1H 7K4 Canada
faisal.qureshi@uoit.ca

***Abstract*—We present a novel technique for image driven shot retrieval in video data. Specifically, given a query image, our method can efficiently pick the video segment containing that image. Video is first divided into shots. Each shot is described using an embedded hidden Markov model (EHMM). The EHMM is trained on GIST-like descriptors of frames in that shot. The trained EHMM computes the likelihood that a query image belongs to the shot. A Support Vector Machine classifier is trained for each EHMM. The classifier provides a yes/no decision given the likelihood value produced by its EHMM. Given a collection of shot models from one or more videos, the proposed technique can efficiently decide whether or not an image belongs to a video by identifying the shot most likely to contain that image. The proposed technique is evaluated on a realistic dataset.**

***Keywords*-Embedded hidden Markov model (EHMM); GIST-like; shot model;**

I. Introduction

Humans have an uncanny ability to recognize whether or not they have seen a video when presented with a frame from that video. We might not be able to exactly identify the location of the frame in a video, yet we can tell if the frame comes from that video. Clearly, we are not remembering (and indexing) each and every frame. Rather, we are constructing a higher order model of the video, which can latter be used to determine if we have seen a particular image in that video. The work presented here is a step towards understanding and replicating this ability. Specifically, we develop a technique for retrieving the video shot most likely to contain a query image. The proposed system then is able to answer the question,

> *I remember seeing this image. Which movie was it, again? Or, I don't think I have seen this image before. Have I?*

The ability to manage, archive and search through large corpus of video data is becoming increasingly relevant. Searching through videos is especially challenging. Here one must first devise a way to pose a query, i.e., there must be some mechanisms through which users can describe what they are looking for in the video. Secondly, there must be a way to execute this query to retrieve the relevant portion from video data or to report a failure when relevant information is not found in the video data. Common schemes typically combine metadata embedded in the video with other cues (image driven, overall narrative structure, etc.) to perform video retrieval [1]. The approach presented in this paper eschews metadata and relies solely upon image cues to retrieve relevant portion from a video. As a consequence the proposed scheme does not support textual queries. Rather the query itself is an image.

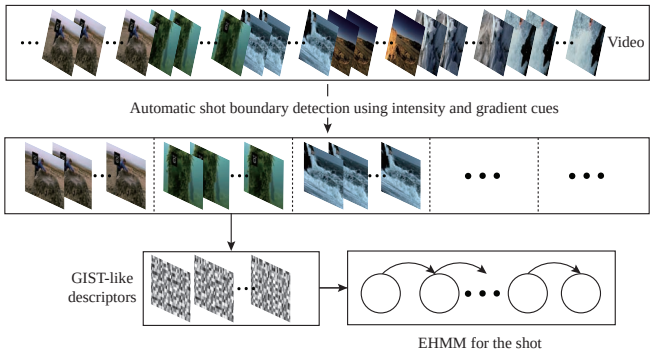
The proposed system first automatically segments the video into *shots*, containing frames having similar visual characteristics. Next each shot is modelled using an embedded hidden Markov model (EHMM). The EHMM is trained by computing GIST-like descriptors for every frame in the shot and constructing observation vectors by sampling the GIST-like descriptor by sliding a 2D discrete Cosine transform (DCT) window (Fig. 1(a)). A Support Vector Machine (SVM) classifier is trained for each EHMM to decide whether or not a particular image belongs to the shot for that EHMM (Fig. 1(b)). While video processing is slow—since it requires 1) shot boundary detection, 2) GIST-like descriptor construction for every frame in the video, 3) training EHMMs for every shot in the video, and 4) learning SVM classifiers for each EHMM—shot retrieval is quick. Shot retrieval requires computing GIST-like descriptor for the query image and running this descriptor through EHMM/SVM for every shot in the video(s). Fig. 1(c) illustrates image retrieval.

II. Overview

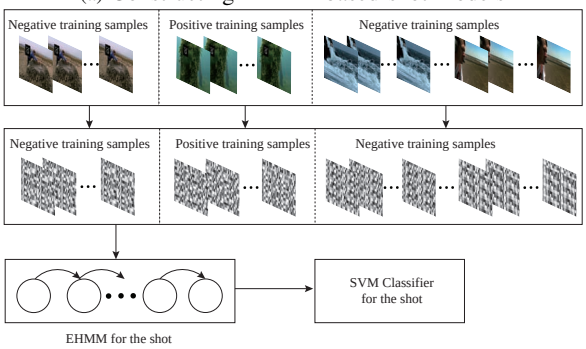
This paper is organized as follows. Sec. III briefly introduces relevant work in video retrieval. We describe video processing in the following section. Shot retrieval is explained in Sec. V and results are presented in Sec. VI. Sec. VII concludes the paper with a brief discussion.

III. Background

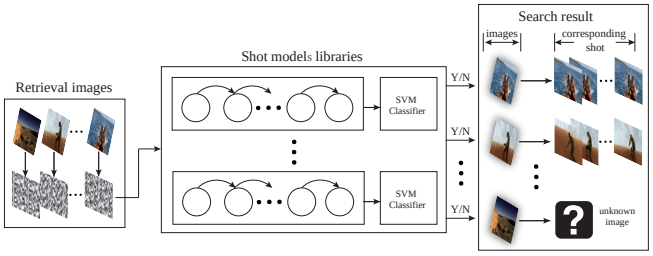
Existing methods that deal with image-based video retrieval typically divide the video into shots [2]. The query image is then matched against the keyframes corresponding



(a) Constructing EHMM-based shot models



(b) Learning SVM classifier for each shot model



(c) Image retrieval

Figure 1: An overview of the proposed system.

to different shots. The challenge here is to identify keyframes that best represent a shot. Shot detection itself is an active area of research. Shot detection aims to partition the video and group frames that share similar characteristics, such as color, motion, texture, etc. Shot detection techniques typically use color and motion cues to detect when two successive frames are sufficiently different to warrant creating a shot boundary.

In the recent years significant attention has been paid to the problem of video retrieval. Researchers have investigated semantic and content based video search. Albanese *et. al.* [3], for example, develop models to achieve semantic mapping between search query and the video. This approach has been employed to detect abnormal behavior in surveillance video footage. Other techniques that track objects and construct activity models for video retrieval applications have also been proposed [4]. While activity based video retrieval strategies are suitable for certain applications, it is not clear how these techniques may support image based video shot retrieval. Primarily because activity can not be discerned from a single image. Additionally, these techniques fair poorly in situations where there is no salient foreground motion. Consider, for example, a scene containing a waterfall, clouds floating above and mist settling down below. It is not immediately obvious how to construct an activity model for this scene.

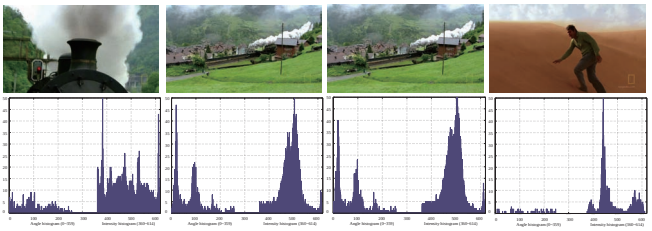
Other techniques take a multi-modal approach to video retrieval. These techniques use multiple cues—such as audio, video transcripts, narrative structure, annotations, etc.—to find relevant segments of a video. Chen *et. al.* [5] proposed a method which uses concept entities to construct the story-structure and semantic relationship between a video and its transcript. Snoek and Worring [6] too proposed a concept-based video retrieval system. Such systems, in general, require manually annotating the video. Manually annotating large corpus of video data is tedious and time consuming. Furthermore, manual annotations are also highly subjective. In contrast the system proposed here does not require manually annotating the video. We refer the gentle reader to [7] for a recent overview of this area.

IV. Video Processing

Each video is first segmented into shots via automatic shot boundary detection. Next one EHMM is trained for each shot. The observation vectors used to train an EHMM are generated by sliding a 4×4 window over the DCT coefficients of the GIST-like features computed for individual frames comprising the shot for that EHMM. EHMMs can then be used to compute the probability that a query image comes from a particular shot. An SVM classifier is trained for each EHMM. Each classifier takes the likelihood computed by its EHMM and returns whether or not the image belongs to the shot corresponding to that EHMM. We describe the various aspects of our technique in the next few sections.

A. Shot Detection

In most cases it is infeasible to test the query image against every frame of a video sequence. An alternative is to collect video frames having similar appearance into (a small set of) different groups, called *shots*. The query frame can then be matched against these shots to determine if a particular shot contains the query image. An efficient scheme for matching a query frame to a shot can yield significant performance boost. During shot boundary detection, each frame is encoded in an orientation+intensity histogram (See Fig. 2). Here intensity values are computed by converting the image into gray-scale; where as, orientation directions are gathered around locations of SURF features [8]. A shot boundary is detected whenever the intensity+orientation



(a) (b) (c) (d)

Figure 2: Intensity and orientation histograms of successive frames are compared using two-sample Kolmogorov-Smirnov test to detect shot boundaries. Frame (top row) are shown along with their intensity+orientation histograms (bottom row). Notice that the histograms for frames (a), (b) and (d) look different since these three frames belong to different shots; whereas, the histograms for frames (b) and (c) look similar since these frames belong to the same shot.

histograms of two successive frames are "significantly" different.

Let $V_{1,n} = \{I_t | t \in [1, n]\}$ denotes a video consisting n frames. Histograms H_t and H_{t+1} of two successive frames I_t and I_{t+1} are compared using two-sample Kolmogorov-Smirnov test, which is defined as

$$D = \sup_x |F_t(x) - F_{t+1}(x)|,$$

where $F_t(x)$ and $F_{t+1}(x)$ are cumulative distribution functions for the two samples H_t and H_{t+1}, respectively. For a given normalized histogram H with m bins, $F(x) = \sum_{i=0}^{x} h(i)$, where $h(i)$ is the i^{th} bin count.

The *null* hypothesis (two frames belong to the same shot) is rejected if

$$\sqrt{n_H^2/2n_H} D$$

is greater than some predefined threshold, creating a shot boundary between t and $t+1$. n_H refers to the total bin count in histogram H_t. Also note that the total bin count in histogram H_{t+1} is n_H since both frames have the same size. Given a set of shot boundaries the original video is segmented into a set of shots $V_{j,k} = \{I_j \cdots I_{k-1}\}$, where $1 \leq j < k \leq n$.

B. Shot Modeling

The next step involves constructing a model for each shot. We propose modeling each shot using EHMM. Given a query image the EHMM returns the likelihood of that the query image belongs to that shot. The EHMM is trained using GIST-like features computed for every frame in the shot.

1) Constructing GIST-like Descriptor: GIST, first appeared in [9], attempts to capture the salient information of the whole image. GIST integrates the information from intensity and color cues at multiple scales and orientations. The latter is extracted by applying a Gabor filterbank to different channels. We have adapted GIST descriptor for our purposes: we use raw RGB values and consider only two pyramid levels when constructing a GIST-like descriptor. Our choice of using only two pyramid levels stems from the fact that in-shot zoom/scale variation is typically restricted to a small range. Fig. 3 shows the original image along with its 16×32 GIST-like descriptor. Below we outline the process for constructing the GIST-like of an image.

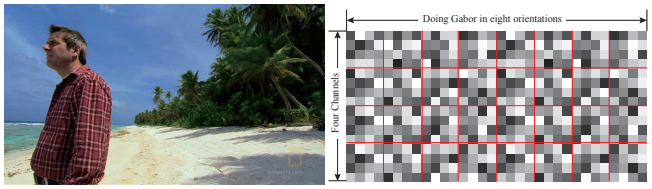


Figure 3: (Left) An image and (Right) its 16×32 GIST-like descriptor. The GIST-like descriptor is scaled to increase visibility.

1: Construct RGB and intensity channels from the input image.
2: **for** Each channel $C \in \{R, G, B, I\}$ **do**
3: Let $C_{CS} = C \ominus \text{interp}(C_{.5})$. Here C_{CS} is the cross-scale center surround difference image for each channel. $C_{.5}$ refers to the second level of the 2-level low-pass pyramid constructed from C.
4: Compute Gabor filter responses $C(\theta)$ of C_{CS} at eight different orientations.
5: **end for**
6: Construct $C_{4\times4}(\theta)$, a 4×4 matrix from each $C(\theta)$. Divide $C(\theta)$ into $4 \times 4 (= 16)$ regions then each pixel of $C_{4\times4}(\theta)$ represents the mean value of the corresponding region.
7: **for all** Orientations **do**
8: Stack $C_{4\times4}(\theta)$ corresponding to R, G, B, I on top of each other to construct $C_{16\times4}(\theta)$ blocks.
9: **end for**
10: Collect these blocks into a 16×32 matrix as shown in Fig. 3(Right).

2) Training Shot EHMM: Fig. 4 describes the EHMM used in our system. It consists of 4 "super" states. An HMM comprising 7 states is embedded within each super state. The super states model the data in the 1st column in the vertical direction (top to bottom); where as, embedded states model the data along the horizontal direction (left to right) in each row. An EHMM models the 2-dimensional data better than an HMM. Furthermore, an EHMM is less computationally expensive than a 2D HMM [10].

From [10], the HMM comprising the super states can be defined as follows:

- N_0, the number of super states;
- $S_0 = \{S_{0,i}\}$, the set of super states;
- $\Pi_0 = \{\pi_{0,i}\}$, initial distribution of super states, where $\pi_{0,i}$ is the probability of being in super state $S_{0,i}$ at time 0; and
- $A_0 = \{a_{i,j}\}$, super state transition probability matrix.

The embedded HMMs are similarly defined as seen below:

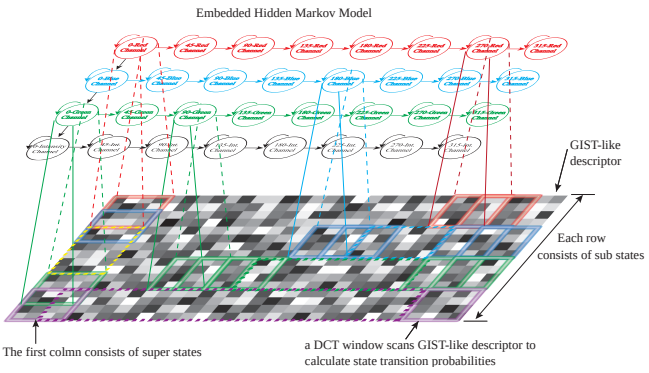


Figure 4: EHMM exploits the 4×8 block-structure of GIST-like descriptor. Each super state refers to one block in the 1^{st} column of the descriptor, where as the 7 sub states corresponds to blocks 2–8 in every row. Observation vectors are constructed by scanning the GIST-like descriptor in horizontal and vertical directions.

- $N_1^{(k)}$, the number of embedded states in the k^{th} super state;
- $S_1^{(k)} = \{S_{1,i}^{(k)}\}$, the set of embedded states in the k^{th} super state;
- $\Pi_1^{(k)} = \{\pi_{1,i}^{(k)}\}$, initial distribution of super states, where $\pi_{1,i}^{(k)}$ is the probability of being in state $S_{1,i}(k)$ at time 0; and
- $A_1^{(k)} = \{a_{i,j}^{(k)}\}$, state transition probability matrix.

Further define state probability matrix $B^{(k)} = \{b_i^{(k)}(O_r, O_c)\}$ for the set of observation vectors at row r, column c. Then, we can represent an EHMM as $\lambda = \{\Pi_0, A_0, \lambda^{(k)}\}$, where $\lambda^{(k)} = \{\Pi_1^{(k)}, A_0^{(k)}, B^{(k)}\}$.

Our EHMM exploits the structure of the GIST-like descriptor computed in the previous section. All states in the EHMM correspond to 1 block in the GIST-like descriptor, e.g., state $S_1^{(1)}$ corresponds to the top-left (4×4) block of the GIST-like descriptor. Observation vectors for the EHMM are constructed by scanning a 4×4 window over the discrete cosine transform (DCT) coefficients of the GIST-like descriptor. The step size for the scanning window is 2. Observation vectors are constructed by scanning the window in both horizontal and vertical directions [10]. The EHMM is trained on all images belonging to the same shot using doubly embedded viterbi algorithm [11].

3) Learning EHMM SVM Classifiers: The EHMM corresponding to a shot computes the likelihood that a query image belongs to that shot. We are interested in a yes/no decision, indicating whether or not a query image belongs to a shot. A naïve approach to get a hard decision given the likelihood value from the EHMM is to use a threshold; i.e., the query image is assumed to belong to the shot if the likelihood returned by the EHMM is above a certain value. This approach is fragile and requires an expert to specify threshold values for every EHMM—a single threshold for every EHMM does not work in practice. Instead the proposed approach learns an SVM classifier to decide whether or not a query image belongs to a shot given the likelihood returned by the EHMM corresponding to that shot. During training an SVM classifier is presented with likelihood values computed by the EHMM along with (+/-) labels, where a + indicates that the likelihood value corresponds to an image that belongs to this shot; a - indicates otherwise. Fig. 1(b) illustrates our approach.

V. Shot Retrieval

Here we describe the shot retrieval procedure. We assume that we are given a set of shot models $\mathcal{S} = \{S_{i,j}\}$, where $S_{i,j}$ refers to EHMM corresponding to shot i in video j. Furthermore, we assume that we also have a set of SVM classifiers $\mathcal{C} = \{C_{i,j}\}$, where $C_{i,j}$ refers to the SVM classifier corresponding to $S_{i,j}$. Given a query image I_q we can construct a set $\mathcal{S}_f$ of shots that might contain the query image as follows:

Require: Set of candidate shots $\mathcal{S}_f = \{\Phi\}$.
1: Construct GIST-like descriptors G_q from I_q.
2: **for all** $S_{i,j} \in \mathcal{S}$ **do**
3: Pass G_q to $S_{i,j}$ and compute likelihood $l_{i,j}$. This is the likelihood of I_q belonging to shot i in video j.
4: **if** $C_{i,j}$ classifies $l_{i,j}$ as $+1$ **then**
5: $\mathcal{S}_f = \mathcal{S}_f \cup (i, j)$
6: **end if**
7: **end for**
8: **if** $|\mathcal{S}_f| > 0$ **then return** "Image seen in these shots ($\mathcal{S}_f$)"
9: **end if**
10: **Return** "Never seen I_q before!"

When needed we can rank shots (i, j) in $\mathcal{S}_f$ according to the likelihoods $l_{i,j}$ to return the shot most likely to contain the query image I_q. Fig. 1(c) depicts the shot retrieval process.

VI. Results

We have evaluated the proposed system on 3 videos, each containing roughly 46 shots. For our dataset each shot consists of approximately 200 frames. For the first set of tests we processed a 30 minutes long video. Processing involved segmenting the video into different shots and learning an EHMM and the associated SVM classifier for each shot. Next we selected 150 frames from 30 different shots. These randomly selected 150 frames served as our query images. Shots corresponding to 120 images were correctly retrieved (80% accuracy). Fig. 5 illustrates this set of tests. A subset of query images are shown in the left column. The middle column shows candidate images selected by the classification stage and the last column shows the keyframe corresponding to the retrieved shot after ranking the set of candidates that survived the classification stage.

For the second test we considered 2 videos. We processed the first video as before. Next we selected roughly 350 frames from these 2 videos and run these query images

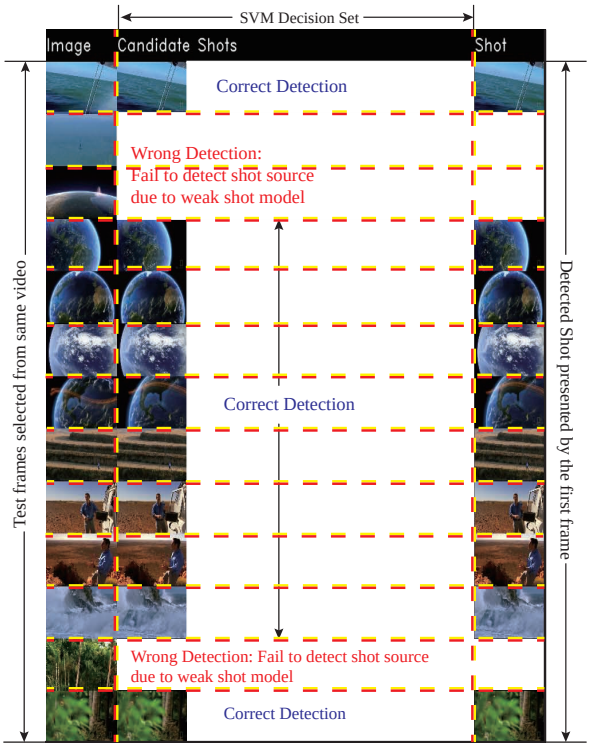


Figure 5: Test 1: A single 30 minute long video is processed for image retrieval. Query images are selected from this video. A selection of query images is shown in the left column. Output of the classification stage is shown in the middle column and the rightmost column shows the keyframes of the retrieved shots.

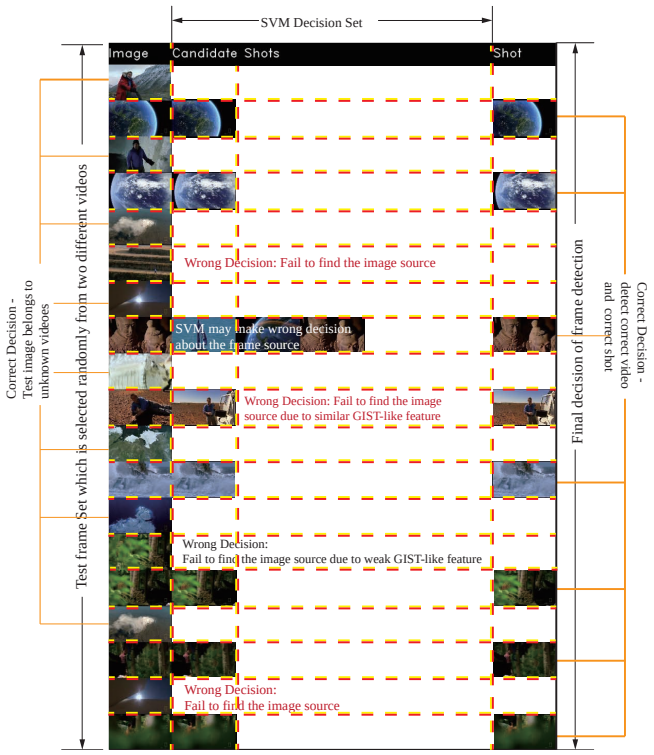


Figure 6: Test 2: Query images are selected from 3 videos (only one of these is processed for image retrieval) are matched against the processed video. A selection of query images is shown in the left column. Output of the classification stage is shown in the middle column and the rightmost column shows the keyframes of the retrieved shots.

Shot Test Result of the Second Test						
			False			
Test Index	Frame Amount	Correct	False Negative	False Positive	Total	Corr. Rate
1	50	44	4	2	6	88%
2	80	72	4	4	8	90%
3	100	77	8	15	23	77%
4	120	103	2	15	17	86%

Table I: Precision and recall results for the second test.

against the processed video. This test allowed us to capture the precision and recall results for the proposed approach. In this context a correct result is defined as follows:

- when a query image belongs to a shot of the processed video and the system is able to retrieve the correct shot for that image; or
- when a query image does not belong to the processed video and the system is able to identify that the query image is previously unseen.

The results are listed in Table. I and Fig. 6 illustrates these tests.

Test 3 (Fig. 7) illustrates that EHMM provides a powerful mechanism for modeling video segments (i.e.,shots). We processed 3 videos as before and randomly selected 100 images from these videos. Here given a query image, we simply rank the likelihoods returned by the EHMMs corresponding to different shots and pick the shot having the highest likelihood, effectively bypassing the classification stage. In this particular test, the proposed scheme achieved 99% accuracy. On a more challenging set of query images the system achieved 80% accuracy.

VII. Conclusions

This paper takes a new look at the problem of image driven video shot retrieval. The proposed scheme relies solely on image cues to find relevant shots in a video given a query image. We are interested in "approximate" search. We are not searching for a frame that matches the query image exactly. Rather we are interested in finding a segment in the video that is most likely to contain the query frame. The system achieves this by training an EHMM for every shot in the video. We have found that EHMM provides a powerful mechanism for describing a shot. It is also interesting to note that a shot containing upward of hundreds of frames

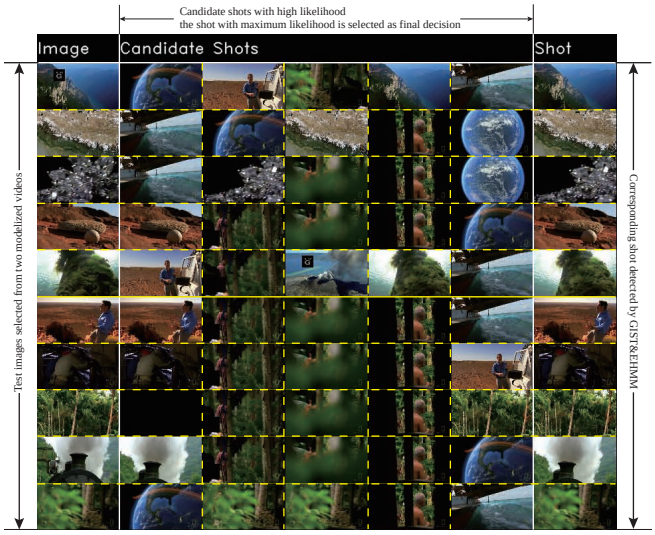


Figure 7: Test 3: 3 videos are processed for image retrieval. Query images are randomly selected from these videos. A selection of query images is shown in the left column. Keyframes for the top 5 ranked shots are shown in the middle column and the rightmost column shows the keyframes of the shots with the highest likelihood.

can be modeled reliably using an EHMM comprising a small number of states. The trained EHMM can later be used for shot retrieval given a query image. We have evaluated the proposed scheme on a realistic, albeit small, set of videos and our results appear promising. We plan to evaluate our scheme on larger datasets in the future.

REFERENCES

[1] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, and G. Quenot, "Trecvid 2011 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proc. of TRECVID 2011*. Geithesburg, US: NIST, USA, December 2011.

[2] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, 1993.

[3] M. Albanese, P. K. Turaga, R. Chellappa, A. Pugliese, and V. S. Subrahmanian, "Semantic video content analysis," in *Video Search and Mining*, 2010, pp. 147–176.

[4] N. Anjum and A. Cavallaro, "Trajectory clustering for scene context learning and outlier detection," in *Video Search and Mining*, 2010, pp. 33–51.

[5] B.-W. Chen, J.-C. Wang, and J.-F. Wang, "A novel video summarization based on mining the story-structure and semantic relations among concept entities," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 295–312, 2009.

[6] C. G. M. Snoek and M. Worring, "Concept-based video retrieval," *Foundations and Trends in Information Retrieval*, vol. 2, no. 4, pp. 215–322, 2009.

[7] W. Hu, N. Xi, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Transactions on Systems, Man, and Cybernetics (Part C: Applications and Reviews)*, vol. 41, no. 6, pp. 791–819, November 2011.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[9] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 300–312, February 2007.

[10] A. V. Nefian and M. H. H. III, "An embedded hmm-based approach for face detection and recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, AZ, USA, March 1999, pp. 3553–3556.

[11] S. shiaw Kuo and O. E. Agazzi, "Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models," *IEEE Transactions in Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 842–848, 1994.