Learning Semantic Visual Concepts from Video

Jingchun Liu and Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside, CA 92521, USA
{liujc, bhanu} @cris.ucr.edu

Abstract

Increasing amounts of digital video data have become available with the rapid growth in video technology. As a result, there is a great need for automatic extraction of concepts or events of interest from video. In this paper, we present an approach for learning concepts from video. The approach consists of three steps. In the first step, video shot boundaries are detected, and from these shots key frames are extracted, which are representatives of the shots. In the second step, key frames are segmented and a variety of features are computed. In the third step, a classification by feature partitioning method is employed for learning different semantic concepts. The results are presented for successfully learning semantic concepts such as ocean, mountain, people, and building from a variety of digital videos.

1. Introduction

With the rapid growth in video technology, more and more information is available as digital video data. Video indexing and retrieval is emerging as an important and challenging problem in multimedia applications. Various features such as color, shape, texture, motion, closed-caption, and speech are being used for retrieving videos. In all of the existing methods, the retrieval is either based on some low-level features or based on examples. From the point of view of users, however, semantic (high-level) concepts are useful and necessary for querying video databases. Therefore, automatically extracting concepts or events in video is a significant requirement for retrieval.

Chang [1] proposed a semantic visual templates method, where templates associate a set of exemplar queries with each semantic concept. The idea is that since a single successful query rarely completely represents the information that the user seeks, it is better to cover the concept using a set of successful queries. This method can achieve good results, but this system has a strong dependence on the users. Naphade and Huang [2] used the concepts of multijects and multinets to represent the semantic features and account for the interaction between concepts. Since the interactions of concepts are vague, they need further guidance to learn the interaction between concepts besides using the multinet to represent relations of concepts. Zhang [3] proposed an object-based video representation method for video compression and retrieval. Lim [4] developed the notion of visual keywords for content-based indexing and retrieval. The idea is to describe a visual document in terms of prototypical visual tokens, visual keywords, and their configuration. All the features are extracted from isolated images, so it lacks temporal information. Existing methods bridge the gap between low-level features and high-level concepts using relevance feedback. For an approach on learning concepts in images based on fuzzy clustering and relevance feedback, see [8].

The problem, which we address, is how to learn semantic concepts from video. Our proposed approach is based on key frames and classification by feature partitioning. Because of the complexity and variability of semantic concepts that can be present in a digital video and great differences between the low-level video features and the high-level concepts, it is not reasonable to apply a small set of features for learning diverse concepts. A natural idea is, therefore, to extract a large number of various image/video features and, hopefully, some of the features can be useful for separating the different concepts. However, a large number of features lead to some problems as well, such as "the curse of dimensionality" and there may be noisy irrelevant features.

In our approach feature selection is carried out during training using a feature partitioning method and weights of features are determined. The relevant features for different concepts are automatically used during the run-time. Our integrated approach extracts various features from video and automatically selects discriminating features for defining a concept. We discuss the concept learning process in an end-to-end system from video shot detection and key frame computation to classification by feature partitioning learning. Experiments are performed to evaluate the capabilities of our approach on real video databases and to compare the classification performance with the commonly used baseline C4.5 algorithm [7].

2. Technical approach

The overall approach for learning a concept from video is shown in Figure 1. In order to learn concepts from video, the first step is to segment the video into a set of basic units called shots. A *shot* consists of one or more frames generated and recorded contiguously, representing a continuous action in time and space. Based on the video

1051-4651/02 $17.00 © 2002 IEEE

shots, we compute the *key frames* from each shot. Using the key frames to index video sequences is obviously far more efficient than using raw video. After that, we perform image segmentation and extract a variety of features based on texture, motion and color. Next we use a classification by feature partitioning method to learn the semantic concept. It performs feature selection to emphasize the most important features that describe the concept. Thus, the learning of a concept is actually based on the features that are associated with key frames in the video clips.

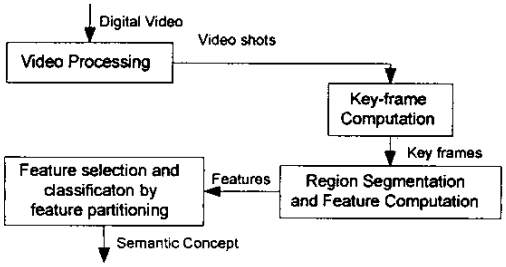


Figure 1. Approach for learning a concept from video.

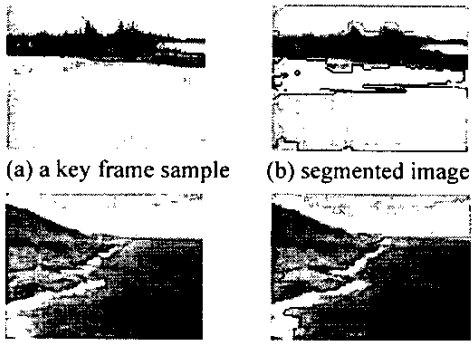
2.1. Video processing and key frame computation

Video segmentation is a prerequisite for structuring and indexing video sources. Since it is not feasible to process all video frames, the features based on key frames are used to learn semantic concepts. In our approach we uniformly and consecutively divide the single frame (still image) into small, non-overlapping square areas, called base windows, denoted by B_{ij} (i,j=0,1,2,...). By estimating the difference of the corresponding regions of successive frames, which consist of base windows, we can detect the significant change between two frames. Then we can decide whether there is shot boundary or not by setting a difference threshold [5]. This method is applied on every consecutive frame in the video sequence.

The shots are usually describe by one or several representative frames, called key frames. We use a *seek and spread* method, based on searching for key frames sequentially [5]. Initially, it compares the first frame to the following ones until finding a different frame or reaching the end of the shot. The frame before the found frame is selected as a key frame. Finally, we extend the representative range of this frame as far as possible. It compares the current key frame to the following frames, also sequentially, until finding a different frame or the end of the shot is reached. Wavelet decomposition is used for computing image similarity.

2.2. Segmentation and feature computation

In order to extract objects from the video key frame, we first segment the image into its three largest regions using the K-means algorithm based on (R, G, B) color features. Figure 2 shows two examples of extracting the three largest regions from key frames. We extract three kinds of features for learning. The global features include texture, means and standard deviations of Gabor coefficients for 4 scales and 2 orientations (feature label 1-16). In addition, the temporal feature, optical flow, is represented in terms of histograms, and sampled uniformly into 8 bins in x and y direction (feature label 17-32). Then we calculate mean and standard deviation values for the whole image, and three biggest regions in R, G, and B color space (feature label 33-56). The same computation is employed in H, S, and V color space (feature label 57-80). So the total number of features is 80.



(a) a key frame sample (b) segmented image

(b) another key frame (d) segmented image

Figure 2. Examples of extracting three largest regions from key frames.

2.3. Feature selection and classification

Based on the features detected, we employ a classification by feature partitioning (CFP) algorithm to learn the concepts in a video. Each key frame is represented as: *<Attributes, Concept> = <Global features, Local features, Concept>*, where *Attributes* consists of global features and local features; and concept is some number that represents the *concept* that we will learn in video. Each feature is treated independently. A feature segment along a feature dimension is the basic unit of representation and it includes lower and upper bounds of the feature values, the associated class, and the number of instances it represents. The class value of a segment may be undetermined.

Initially, a feature segment covers the entire range of a feature dimension, that is, $\{(-\infty, +\infty)$, *undetermined*, $0\}$. Here, the first element of the triple indicates the range of the segment with lower and upper limits, the second its class, and the third, called the representative value, is the number of examples represented by the segment. For each feature vector, we project the N-D feature space onto each feature dimension and form the clusters corresponding to different classes. In order to classify key frames we need to estimate the degree of similarity to every class over different feature dimensions. It can be represented as

$$Simi(F,C_i) = \sum_{j=1}^{N}(w_j \cdot Sign(f_j))$$, where M is the number of classes; N is the number of features; $F = (f_1, f_2, ..., f_N)$ is feature vector of a key frame; w_j is the feature

weight; C_i is the *i-th* class, $i=1,...,M$; and *Sign* is a function of a feature value, which can be represented as:

Sign =1 if f_j projects to a segment belonging to class j; otherwise *Sign* =0.

Finally, we choose the class with the maximum similarity from all classes by,

$$C = \max_i \left(Simi\left(F, C_i\right)\right),\ i = 1,2,...,M$$

In order to generalize (i.e., extend) a segment in feature f_j to cover a point, the distance between them must be less than a given generalization limit (D_{fj}). Otherwise, the new example is stored as another point segment in the feature dimension f_j. If the feature value of a training example falls in a segment of the same class, then the representative value is incremented by one. On the other hand, if the new training example falls on a segment with a different class than that of the example, CFP specializes the existing segment by dividing it into two sub-segments and inserting a point segment (corresponding to the new example) in between them. When a segment is divided into two segments, CFP distributes the representative value of the old segment among the new ones in proportion to their sizes.

The training process in the CFP algorithm has two steps: learning the feature weights and learning the feature partitions (Figure 3). The set of training instances, global weight adjustment rate (Δ), and the vector of generalization limits (D_f) are the arguments of the training procedure. For each training instance, the prediction based on a feature is compared with the actual class of the example. If the prediction by a feature f is correct, then the weight of that feature, w_f, is incremented by $w_f \cdot \Delta$; otherwise, it is decremented by the same amount.

The process of classification is described in Figure 4, where $Vote_c$ denotes a similarity prediction made by each feature and f_j denotes the representative value. The classification depends on a vote taken among the predictions made by each feature. The effect of the prediction of each feature in the voting is based on the weight of that feature. The predicted class of a given instance is the one, which receives the highest amount of votes among all feature predictions. During the classification, the initial classification by feature partitioning algorithm just gives the unknown label to the testing instance. If the number of training examples is limited, it will cause segments, which belong to the same class, separated from each other. In this situation we use *1*-Nearest Neighbour algorithm to solve this problem and give a class label to the instance for unknown segments. Using the votes from different classes, we define a confidence metric, which can be used to estimate degree of overlap among different concepts.

$$Confi(k) = \arg\max_k \left(\sum_{i=1}^{N} vote(k, f_i) \right) / \sum_{k=1}^{M} \sum_{i=1}^{N} vote(k, f)$$

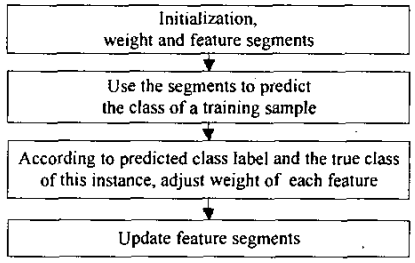


Figure 3. Block diagram for training.

Comment: a test instance is vector $f=[f_1 f_2, ... f_N]$
Classification (f):
Initialization, for each class $_c$, $Vote_c$ =0
For each feature dimension f_j
If f_j value is known
c=*predition_on_feature*(f_j)
If $c \neq$ undetermined
$Vote_c = Vote_c + w_f$
Else
Classify using 1-NN;
$Vote_c = Vote_c + w_f$
Return class c with the highest $Vote_c$

Figure 4. Classification algorithm.

2.4. Parameter optimization using GA

During the training process, we need specify the N generalization limits for all features and one weight adjustment rate. These N+1 parameters can be estimated using the genetic algorithm (GA). The chromosome is a real-valued vector representing N+1 domain parameters. We employ crossover, mutation, and reproduction operators to select the chromosome with the higher fitness value. We obtain the classification precision for each chromosome. This normalized precision is used for fitness value.

In setting the initial population, the Δ values are randomly chosen from [0,0.1], and the D_f values are randomly chosen from [0.0005,0.02]. All feature values are normalized to the range [0,1]. The population size is 200, the crossover rate is 0.5, and the mutation rate is 0.2.

3. Experiment results

We use MPEG video data to demonstrate our proposed approach for learning concepts. We input different key frames extracted from video shots, and learn concepts "ocean", "mountain", "people" and "building". Table 1, which gives details of the data, shows that a total of 588 key frames are obtained automatically from 68582 video frames. Figure 5 shows examples of key frames used for learning the concepts. Out of 588 key frames, some key frames do not belong to any known class of four concepts considered here. We remove these key frames and choose only 400 key frames for training and testing. This allows us to perform proper training and evaluate performance of

Table 1. The video information from some databases.

Videos	# of Frames	# of Shots	# of Key frames
Bahamas	10179	57	90
CarinsA	16115	64	97
CarinsB	24539	97	162
HawaiiA	2796	17	20
HawaiiB	7063	29	46
HawaiiC	7890	33	47
HawaiiD	16780	84	126
Total	68582	381	588

Table 2. The experiment results for semantic concept detection from video.

Concept	Training sets (positive, negative)	Testing sets (positive, negative)	Detection Accuracy	False Alarm
Ocean	(61, 139)	(61,139)	74%	17%
Mountain	(39, 161)	(38,162)	61%	1%
People	(70, 130)	(72,128)	81%	20%
Building	(30,170)	(29,171)	69%	1%

Table 3. Confusion matrix for learning concepts.

Class / Test	Ocean	Moutain	People	Building
Ocean (61)	45	0	14	2
Mountain (38)	9	23	6	0
People (72)	12	2	58	0
Building (29)	3	0	6	20

Table 4. Comparison of CFP and C4.5.

Comparison of detection accuracy	CFP	C4.5
200 training; 200 testing	73%	54%

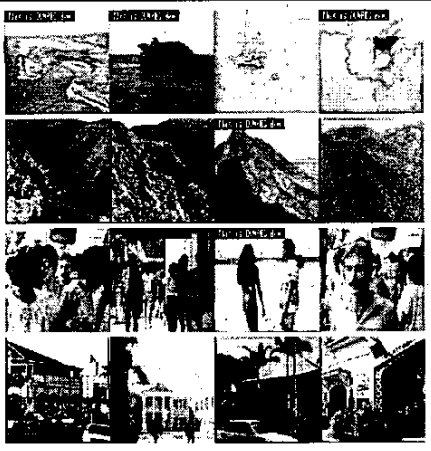


Figure 5. Examples of key frames for learning.

CFP accurately without the unknown class. Half of the data is used for training, and the other half for testing. The performances of our system on 200 test data are shown in Table 2 and Table 3. In Table 2 we show negative set to obtain detection accuracy and false alarm. From the experimental results we can see that the system can learn four different concepts "Ocean", "Mountain", "People" and "Building" efficiently. The detection accuracy varies from 61% to 81%, while the false alarm varies from 1% to 17%. Examining the confusion matrix in Table 3 and typical key frames in Figure 5, we see that "Ocean" is classified as "People" (14/61) because one key frame may have multiple concepts and we just consider the dominant concept. We compare the CFP algorithm with another important learning algorithm, C4.5 [7]. The comparison of CFP and C4.5 is given in Table 4. The average detection accuracy of CFP is 73%, which is significantly better than 54% accuracy of the C4.5 algorithm.

4. Conclusions

In this paper, we present an approach for semantic concept learning from video data. The success of this key frame-based approach depends on a large set of features and a powerful technique for feature selection for learning concepts. The classification by feature partitioning approach is computationally efficient; it assumes the features are independent and thus, ignores correlation among features (as a result there maybe a slight degradation in performance). In the future, we will extend our system to learning spatio-temporal concepts that involve both spatial and temporal relations.

Acknowledgement: This work was supported in part by the grants F49620-97-1-0184 and DAAD19-01-0357; the contents and information do not necessarily reflect the position or policy of U. S. Government.

References

[1] S.F. Chang, W. Chen, and H. Sundrarm, "Semantic visual templates-linking features to semantics," *Proc. 5th IEEE Int. Conf. on Image Processing*, vol. 3, pp. 531-535, Chicago, IL, Oct 1998.

[2] M.R. Naphade and T.S. Huang, "A probabilistic framework for semantic video indexing, filtering and retrieval," *IEEE Trans. On Multimedia, 3(1), pp. 141-151, March 2001.*

[3] H.J. Zhang, J.Y.A. Wang and Y. Altunbasak, "Content-based video retrieval and compression: A unified solution," *Proc. Int. Conf. on Image Processing*, vol. 1, p.13-16, 1997.

[4] J.H. Lim, "Learning visual keywords for content-based retrieval," *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, vol. 2, pp. 169-173, 1999.

[5] W. Xiong, J.C.M. Lee and R. Ma, "Automatic video data structuring through shot partitioning and key-frame computing," *Machine Vision and Application*, vol. 10, pp. 51-65, 1997.

[6] H.A. Guvenir and I. Sirin, "Classification by feature partitioning," *Machine Learning*, 23, pp. 47-67, 1996.

[7] J.R. Quinlan, "C4.5: Programs for machine learning," *Morgan Kaufmann Publishers*, 1993.

[8] B. Bhanu and A. Dong, "Concepts learning with fuzzy clustering and relevance feedback", *Engineering Applications of Artificial Intelligence*, 2002.