Automatic Extraction of Highlights from a Cricket Video using HMM and MPEG-7 Descriptors

P. Chappidi, K.R. Kothinti, and K.R. Namuduri

Department of Electrical and Computer Engineering, College of Engineering

1. Introduction

Sports videos are usually lengthy and they appeal to a vast crowd. Though a sports video is lengthy, most of the viewers prefer to watch particular segments of the video which are interesting, like a home-run in a baseball or goal in soccer. When compared to the total length of the video, these segments form only a small portion. Hence these videos need to be summarized for effective data management and presentation. Detection of important events and summarizing a video makes it possible to deliver sports video over narrow band networks, such as internet.

2. Overall Approach

Our approach is based on the observations, that a highlight has a certain pattern of transition between different shots.

We identified four types of shots, they are, 1) pitch view, 2) close-up view, 3) audience view and 4) medium view. For instance, a boundary can be composed of pitch view followed by a medium view which is followed by audience view and a close-up view. A shot is classified into any of the above shots by the features it has. These features include the amount of color, texture, motion and number of edges. In our approach a Hidden Markov Model (HMM) is designed to model the highlight. Every shot and its transition to the next shot are compared with the model. The shots that have same transition pattern as that of the HMM are extracted to form the highlight.

Extraction of features for classification of shots

A video is a combination of different “scenes”. A “scene”, in turn is a combination of “shots”. And a “shot” is continuous recording from a single camera. A shot by itself does not make any sense. But when different shots are combined together, by “editing”, they make sense. Features like color, texture and motion, which are very significant and can be computed efficiently and reliably in an image are extracted in order to identify the view. The video of interest is first divided into shots. This division is done by using the MPEG-7 color histogram. All the frames in a shot have same kind of distribution of features. The frames in a single shot which have slight variations in color are called the key frames of the shot, and the number of key frames in a shot can be one or two or three, depending on the number of frames and variations in the shot. In order to reduce computation time, all computations are done on the key frames. For motion computation, other frames of the shot are used as motion cannot be calculated using two or three frames.

The main features that we are focusing on are, 1) grass color, 2) pitch color, 3) texture, 4) motion and 5) number of edges. A view can be classified into any one of the four views:

Pitch view

In a pitch view, the amount of pitch color (sand color) present is more. Although other features like grass color is also present, but in small amount when compared to pitch color.

Medium view

In a medium view, the amount of grass present is more. Grass color is dominant with respect to pitch color.

Close-up view

In a close-up view, the amount of motion is less, as a close-up shot is used to show player.

Audience view

In this view audience are shown. Audience view usually has many edges; an edge here means significant change in the color, and a specific type of texture. So, these features can be used to classify this view.

Color value extraction:

For color value, MPEG-7's Dominant Color Descriptor (DCD) in RGB color space is used.

Texture value extraction:

For texture computation, MPEG-7's Homogeneous Texture Descriptor (HTD) is used.

Motion value extraction:

Motion is calculated between the frames in a shot by finding the MSE between two blocks located at same position in the frames. The variance of these MSE values is computed and depending on the variance value, intensity of motion can be known.

Edge value extraction:

Number of edges in an image is found out using the “Canny" edge detection method.

The feature values are calculated for all the key frames in a shot and average of all these values is computed. Depending on these values of features we can estimate which shot it is. In order to apply these values to HMM, we need to know the following.

Elements and three problems of HMM

- ***N,*** Number of states. Each view is considered to be a state. Therefore there are four states.
- ***M***, Number of distinct observation symbols per state. Features that we extract are the observations. For each state we have five observations.
- $A = \{a_{ij}\}$, State transition matrix. Each element of the matrix gives the probability of transition from one state to another.
- $B = \{b_j(k)\}$, Observation symbol probability. Which gives the probability of being in state *j* and observing symbol *k*.
- $\prod = \{\pi_i\}$, Initial state probability. Which gives the probability of a state being the first state.

In compact form the HMM model is represented as $\lambda = [A, B, \pi]$.

Problem 1: Given observation sequence *O* and model λ, how to compute $P[O \mid \lambda]$, i.e. probability that the observed sequence is produced by the model.

Problem 2: Given model λ and observation sequence *O*, how to choose the best state sequence *Q*.

Problem 3: How can we adjust the model parameters, $\lambda = [A, B, \pi]$ in order to maximize $P[O \mid \lambda]$.

In automatic extraction of highlights, the views are not known, but we know the observations which are the features that we compute. We need to identify the type of view from the features. Let V_i, i = 1, ..., 4, be any one of the four views of interest. O_k, k=1, …, 5, be the observations from each view. We need to find the probability of a view given the observations, which can be given as $P(V_i \mid O_k)$. This can be calculated using the Bayesian rule

$$P(V_i \mid O_k) = \frac{P(O_k \mid V_i) P(V_i)}{P(O_k)} \qquad (1)$$

We have to find the best state sequence which follows the state transition pattern as our own model. These states can be extracted from the video to form the highlights. In order to find the best state sequence that follows the state transition pattern as our model, we use the Viterbi Algorithm. For this we define a variable called forward variable $\alpha t(i)$, t = 1, …, N & i = 1, …, 4, where N is the number of shots. α values in the first time instance are computed as

$$\alpha t(i) = \pi_i * V_i \qquad (2)$$

where *t=1* and number of states i = 1, ..., 4, π values are estimated by observations. V_i is the probability of a view, computed using (1). Iteratively α values from time instance 2 are calculated as

$$\alpha n(i) = [\alpha n-1(1)a1i + \alpha n-1(2)a2i + \alpha n-1(3)a3i + \alpha n-1(4)a4i] * Vi \qquad (3)$$

where *n = 2, ..., N* and number of states *i = 1, ..., 4,* V_i is the probability of a view and a_{ij} are the state transition probabilities. Equation (3) gives how a state *i* can be reached at time *t+1* from all the 4 possible states at time *t*. In order to find the best state sequence, we need to find the maximum α value at each time instance.

3. Conclusions

Automatic video summaries are very useful and important tools. Highlights are useful for various purposes like, to see exciting events of a game, also highlights can be used for developing strategy against a team or improving a teams performance. Since the highlights are small in size, they can be made available to a vast crowd via internet and also they occupy less storage space.

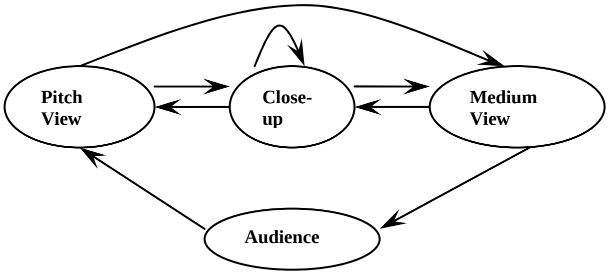

Figure 1. HMM model for highlights

4. References

1. B.S Manjunath, J.R.Ohm, V.V Vasudevan and A.Yamada, “Color and texture descriptors," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, 2001, pp. 703-715.
2. L. Rabiner, “A Tutorial on Hidden Markov Models ansd Selected Applications in Speech Recognition, "Proceedings of the IEEE, Vol. 77, No. 2, Feb 1989.
3. Peng Chang; Mei Han; Yihong Gong, “Extract highlights from baseball game video with hidden Markov models”, Page(s): I-609- I-612 vol.1.