Text Driven Temporal Segmentation of Cricket Videos

Pramod Sankar K., Saurabh Pandey, and C.V. Jawahar

Centre for Visual Information Technology,
International Institute of Information Technology, Hyderabad, India
jawahar@iiit.ac.in

Abstract. In this paper we address the problem of temporal segmentation of videos. We present a multi-modal approach where clues from different information sources are merged to perform the segmentation. Specifically, we segment videos based on textual descriptions or commentaries of the action in the video. Such a parallel information is available for cricket videos, a class of videos where visual feature based (*bottom-up*) scene segmentation algorithms generally fail, due to lack of visual dissimilarity across space and time. With additional *top-down* information from textual domain, these ambiguities could be resolved to a large extent. The video is segmented to meaningful entities or scenes, using the scene level descriptions provided by the commentary. These segments can then be automatically annotated with the respective descriptions. This allows for a semantic access and retrieval of video segments, which is difficult to obtain from existing visual feature based approaches. We also present techniques for automatic highlight generation using our scheme.

1 Introduction

The significance and challenge of temporal segmentation of videos into meaningful entities, is parallelled only by its spatial counterpart. Much of the previous work in video segmentation has focused on shot-cut detection. Contiguous frames in the video, which have little change in visual content are generally grouped into a *video shot*. A shot change or a *cut* is detected, whenever the camera shifts, or the scene being captured changes significantly. However, our work focuses on obtaining a *scene segmentation*, which is a meaningful entity of a video [1]. This work is motivated by the following facts:

- Shot-Cut detection, using visual features, has been well addressed in literature [2,3,4]. However, the video shot obtained from cut detection is not generally a meaningful entity. Shots are a low-level or syntactic representation of the video content, while for the purpose of recognition, annotation and retrieval, a higher level semantic representation such as a "scene", is required.
- Semantic access to content, has met with much success in the text retrieval domain. Much work exists on mining and retrieving semantic concepts from document collections.
- A parallel text is available for many videos such as closed captions for news videos, subtitles for movies, lyrics for music videos, commentary for sports videos, etc. This text is a reliable source of information regarding the content of the video.

P. Kalra and S. Peleg (Eds.): ICVGIP 2006, LNCS 4338, pp. 433–444, 2006.

- With a synchronisation between the text and the video, the video segments would correspond to a textual description of the video. This allows for automatic annotation of video segments with the associated text. The videos could then be accessed at the semantic level and retrieved using human-understandable textual queries.

Segmenting a video into meaningful entities is very challenging since there is a lack of correspondence between the meaning of the scene and the visual features. Previous work that segments a video into scenes [1,5] using visual features [6,7] or scene dynamism [8], fail in many cases where there is no significant visual change across space and time. This is especially true for the class of sports videos. However, this class of videos have the advantage of being associated with a textual description in the form of a commentary that is generally available in parallel. This text provides ample information regarding the scene content and where and how it changes.

The Problem: In this paper we address the problem of segmenting a video into meaningful scenes, using the text that describes the video. Specifically, we use the commentaries available for sports videos, to segment a cricket video into its constituent scenes, called *balls*. Once segmented, the video could be automatically annotated by the text for higher-level content access.

The Challenges: The scene changes in a sports video are highly ambiguous, since there is no fixed point where one event ends and another begins. The videos are characterised by diverse visuals within the scene and very similar visuals across scenes (at the scene boundaries). This makes it difficult to find scene changes, using purely visual domain techniques. To complicate things further, during broadcast, a large number of *replays* are shown, which are not synchronous with the flow of the match. Moreover, the broadcast contains a large number of scenes, videos, graphics etc. that closely resemble the actual match. They also contain a large number of advertisements that overlap in visual content with the match scenes.

Apriori Knowledge Used: The ambiguities in the visual domain could be resolved by using parallel information for the video. The parallel information could be obtained form two sources: i) audio and ii) text. The audio in a sports video would consist of the commentators' running commentary and the audiences' reaction. The audio is available only in a feature space, which needs to be converted to a more meaningful domain (such as text) by using various speech recognition modules (which are inherently complex and not totally accurate). Moreover the information from the audio domain is as ambiguous as the visual domain (for very similar reasons). On the other hand, textual commentaries, as available from websites such as Cricinfo.com(TM), are meaningful, reliable, accurate, and complete, with regards to conveying the proceedings of the event. Textual descriptions are accurate and meaningful, and immediately correspond to a semantic representation. The semantics provide clues regarding the visual content, when described using visual scene categories. By identifying the scene category from text, the visual content could be estimated.

The Semantic Gap: The top-down information from text and the bottom-up information from visual features has to be synchronized and merged. The top-down information defines the approximate *content* of a video segment and the bottom-up techniques

should be used to segment the video such that it *appears* similar to the model defined for the segment. However, the commentaries are a high level conceptual description of the scene, which cannot be directly represented using visual domain features, the so-called *Semantic Gap*. A mechanism is required to bridge the semantic gap by finding correspondences between scene changes and the scene descriptions. This is achieved by building approximate scene models for each of the scene categories.

In other words, the top-down (textual) and bottom-up (visual) information needs to be effectively merged to solve the problem on hand (segmentation). The visual clues are used to estimate a scene segmentation, which is refined by constraining the segmentation to look similar to the model defined. The optimization of this estimation is performed using the Maximum Likelihood (ML) framework. Though explained in the context of cricket videos, our techniques can be directly extended to any class of videos where the events occur from a given set of categories.

Automatic Annotation: Following the segmentation, the textual description is directly synchronized with the video segments. Thus, the segments could be automatically annotated. Automatic annotation of multimedia is of immense interest in the information retrieval community. Existing content based retrieval systems are computationally expensive and few approaches can robustly retrieve from large video collections. Text annotations of video allow us to build a text based retrieval system for videos, which is very quick and efficient.

2 Visual Domain Processing of Videos

It is common to use the domain knowledge of the class of videos for processing them, such as [9] for baseball, [10] for American football, [11] for tennis, and [12,13] for cricket etc. We use the domain knowledge of the videos to build scene categories and approximate scene models. The scene in cricket, is called the "ball" (similar to a "pitch" in baseball). The ball is defined to begin with the bowler running to deliver the ball, and end at the start of either i) the next ball, ii) a replay or iii) an advertisement. A ball consists of the bowler running to deliver the ball, the ball being delivered, played, fielded and returned. There are a minimum of six balls per *over*, and 50 overs for each side to play. Between consecutive overs there is a lengthy break which is typically filled with advertisements in the broadcast. A large number of replays are generally shown between the balls. A conceptual description of the broadcast cricket video is given in Figure 1.

In this section we describe the visual domain processing of the videos. We first detect shot changes in the video and categorise the shots into one of several classes. These shot classes and shot class transitions are used to model the scene categories as described in Section 3.

Shot Detection. Much work exists in shot-cut detection [2,3]. Popular techniques that use image features [7], optical flow [14] etc., are not applicable due to the heavy noise that is common in broadcast videos. In such cases, a histogram based descriptor is well suited [4]. To ensure invariance to minor changes in RGB values and to noise, the RGB axes are binned, and each pixel is assigned to the cube that the bins describe. To enforce

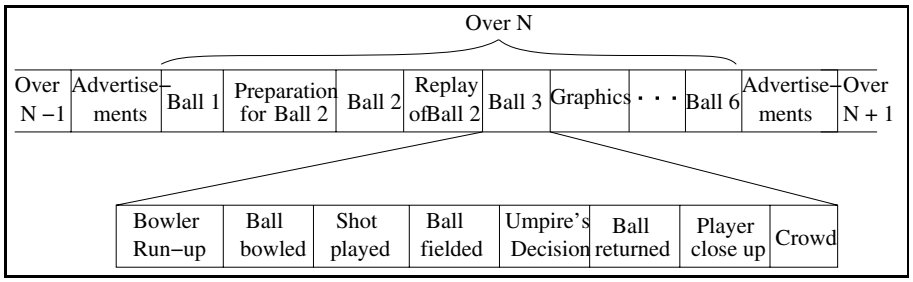


Fig. 1. Depiction of a generic cricket video. Each *over* has 6 (or more) *balls*, each scene consisting of the ball being delivered, played, fielded and returned. In a broadcast, replays and graphics are shown between the scenes and advertisements between the overs.

spatial consistence, we divide the frame to N blocks and build the binned histograms for each block. For cut detection, the histograms of consecutive frames are compared and a cut is detected if the difference is above a particular threshold. The threshold for the given video is found using the technique described in [15].

Soft Classification of Shots. The detected shots are classified into one of the shot categories. For cricket videos, these are the set $C = \{$*pitch view, run-up view, player close-up, crowd, advertisement, replay*$\}$. The different shot classes are shown in Figure 2. Though these classes exhibit wide disparity over different matches, the features from the video for a given cricket match (or in many cases a given tournament of matches) are very similar. The representative histogram feature vector $C_i = f_{C_i1}, f_{C_i2}, ..., f_{C_in}$ for each shot class is learnt from training data. Each shot $S = f_{S1}, f_{S2}, ..., f_{Sn}$ is compared with the class-representative feature vector C_i, using the L1-Norm to obtain $d(S, C_i) = \sum_{k=1}^{n}(f_{C_ik} - f_{Sk})$. The shots are classified using the maximum likelihood estimate as

$$Class(f_{S1}, f_{S2}, ..., f_{Sn}) = arg\max_i \frac{d(S, C_i)}{\sum_k (d(S, C_k))}$$

The accuracy of shot classification is presented in Figure 3 (a).

Another class of shots that we need to handle are the advertisements and replays. Previous advertisement detection methods [16] rely on the intensity in activity from the large variations in the video frames. However, this is also valid for action sequences in a sports video. Replay detection techniques [17] have used a replay transition detection, or slow motion as a clue, which are not applicable for our case. Instead, it was observed that for advertisements and replays, the video production removes the scoreboard at the bottom, that is generally present for the match play, as can be seen in Figure 2. The scoreboard could be detected to distinguish between match play and advertisement/replay. Our method provides a detection accuracy of 82.44% for the class advertisemnts/replays.

Segmenting using Visual Features. Scene segmentation in visual domain could be performed by using the pitch views as *canonical* scenes [18] that bound the action. However, due to the large number of replays, and the inaccuracy of shot classification, the

Fig. 2. Example frames from the shot classes, from left to right: Ground view, West Indies player, Crowd, Pitch view, Indian player and Advertisement. Note that the scoreboard present in the bottom of the screen for the shot classes, is absent for the advertisement.

identified pitch views are more than the number of balls, consequently, yielding poor segmentation. It was observed that over a duration of 8 hours of a match with 629 balls, 945 segments were obtained, where the extra segments come from repeated pitch view shots. Moreover, a large number of balls (52) were missed due to inaccurate shot classification. By enforcing a minimum time duration for each segment, a large number of false positives were eliminated, but many outliers still remained. Also the segmentation tends to favour segments of the same size, while the duration of the scenes would actually depend on the scene category.

3 Modelling the Scene Categories

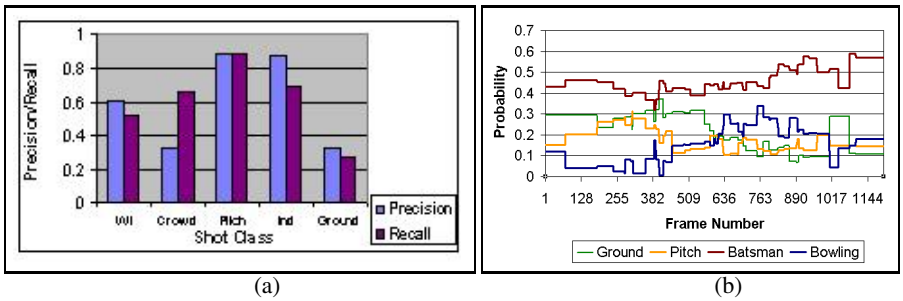
In cases where the visual domain techniques are insufficient for scene segmentation, a parallel textual description could be used to provide additional information. Such parallel text is available for sports videos in the form of online commentaries. For. eg., the commentary of a cricket game is given below:

13.1 Smith to Sehwag, FOUR, short of a good length and outside the off, driven on the up superbly through cover, the timing and placement are excellent, Bravo dives desperately but can't quite pull it back

13.2 Smith to Sehwag, 1 run, played away for a single

13.3 Smith to Yuvraj Singh, FOUR, short of a length and outside the off, Yuvraj stands tall and times that magnificently through cover point. That is a good shot from Yuvraj!

It can be seen that the commentaries contain heavy usage of the domain specific vocabulary, which is a highly conceptual representation. Mapping such semantic concepts



(a) (b)

Fig. 3. (a) Precision-Recall of the shot classification (b) Scene model for the outcome FOUR

to lower level features from images/video is a major challenge corresponding to bridging the *semantic gap*. We model the scenes by finding an approximate map between the shot and scene classes. We assume that each scene class could be characterized by the shot classes it consists of, along with their durations and transitions. The scene building is described in Algorithm 1. The input to build the scene model are the training example scenes $< V_1, V_2, ..., V_n >$ of the scene S_s. Let L_{V_i} be the length of the video V_i. The text commentary is used to generate a hypothetical video's representation that is used as the model for the entire match.

The average duration of a given scene is computed from the examples and used to build a descriptor for the scene. The scene descriptor is the set of probabilities for a given frame to belong to each of the shot classes. These probabilities model the scene to a large extent. For example, in case of the outcome *four*, the pitch view would generally be followed by the ground view for some time, and the camera would then shift to the players. Such probabilities are computed for each frame and normalized to the average length of the scene. The scene model for the outcome FOUR is shown in Figure 3(b).

Algorithm 1. Train_Model(S_s, $< V_1, V_2, ..., V_n >$)

1: Find average length L_{Si} of the videos $< V_1, V_2, ..., V_n >$
2: Set S_s = NULL
3: **for** V_i = V_1 to V_n **do**
4: Identify shots, $C_{i_1}, C_{i_2}, ..., C_{i_m}$ in V_i
5: **for** each shot j = 1 to m and each shot-class k = 1 to l **do**
6: Find probability $P_k(C_{i_j})$, of shot C_{i_j} belonging to the k th shot-class
7: **end for**
/*Build scene representation S_{s_i} as */
8: **for** j = 1 to L_{V_i}, and each shot-class k = 1 to l **do**
9: Append $P_k(C_{i_j})$ to $S_{s_{i_k}}$
10: **end for**
11: Scale $S_{s_{i_k}}$ to average length L_{Si}
12: For each k = 1 to l, Append $S_{s_{i_k}}$ to S_{s_i}
13: **end for**
14: Average S_{s_i} over i = 1 to n to obtain S_s
15: **return** S_s

The above representation builds a model for the intra scene shot changes. To describe the video completely, an inter-scene model is required. The inter scene model describes the probabilities of a particular scene following a given scene. This is modelled for the purpose of handling advertisements and replays. The model learns the probability of an advertisement or replay to follow a given scene. Generally a replay follows a scene belonging to *four* or *six* etc., and advertisements follow an *out* scene. The intra-scene and inter-scene models are used to provide a model for the video to be matched against.

4 Text Driven Segmentation of the Video

4.1 Maximum Likelihood Formulation of Scene Segmentation

The segmentation procedure should identify the begin and end frames of the scenes, over the entire video. The real scene boundary Z_i is assumed to be fixed but unknown. The estimate z_i of the scene boundary, is assumed to be found near the real boundary Z_i with a probability distribution that follows a Gaussian. The Gaussian is centered around Z_i, with a variance σ. The estimate z_i is obtained from visual-temporal information. Let such an observation of the beginning and end of a scene S_i be z_{i_1} and z_{i_2} respectively. The likelihood that shot S_i bounded by z_{i_1} and z_{i_2} actually corresponds to a real scene X is given by $P(S_i|X) = P(z_{i_1}, z_{i_2}|X)$. This likelihood corresponds to a local cost of corresponding S_i to X. The global cost of matching scene estimate set γ with real scene boundaries is given by

$$L(\gamma) = p(Z_1, Z_2|\gamma) = \prod_{0<i<n} P(z_{i_1}, z_{i_2}|X)$$

where n is the number of shots in the video. The maximization of the global likelihood function corresponds to minimizing its negative logarithm. In cases where the scenes are not represented by a known model, the optimization of this function could be done using an Expectation Maximization approach, where both the segmentation and scene parameters are learnt simultaneously. However, using the textual information, the appropriate scene models could be plugged into the likelihood computation. The minimization in such a situation would correspond to a simple weighted matching or assignment problem, which could be solved in polynomial time using dynamic programming. The derivation of local cost between a scene estimate and a scene model is derived following the building of scene models in Section 4.2.

The Generative Video Model. In an ML framework, the observed data, D, or the given video, needs to be compared with an assumed model for the data M. In a general model fitting problem, there are two unknowns: i) the model parameters and ii) the mapping of the data to the model. These unknowns are estimated using an Expectation Maximization procedure. This would be a bottom-up approach. The results of a purely bottom up approach would be poor, due to the ambiguities present in the observed data and the absence of an appropriate scene model. Top-down information in the form of textual descriptions, could be used to identify the scene models and parameters. With such

Algorithm 2. Generate_Video_Representation($Match\_Commentary$)

1: Set G = NULL
2: Parse $Match\_Commentary$ and identify the ball B_i and their corresponding Outcomes O_i
3: **for** each ball i = 1 to n **do**
4: Identify scene model S_s for O_i
5: Append S_s to G
6: **end for**
7: Return G

information, the only unknown that remains, is the mapping of the observed data to the assumed model. The model M is built from the $Match\_Commentary$ using Algorithm 2.

4.2 Segmenting Using the Video Model

The model M, would be a hypothetical video, generated from the scene descriptions. For each ball in the match, the scene category is identified and the corresponding scene model is appended to the generated video. Advertisements and replay shots are added based on the probability of their occurrence following a given scene. The generated model provides an approximation of the shot and scene changes in the video for the given scene description. The mapping of D to M, can be computed using a Dynamic Programming (DP) technique [19]. Assuming that the distance array in the DP procedure is given as D, we use the DP cost computation:

$$D(i,j) = \min \begin{cases} D(i-1,j) + c(i,0) \\ D(i-1,j-1) + d(i,j) \\ D(i,j-1) + c(0,j) \end{cases}$$

where the local distance between two frames, i and j is given by

$$d(i,j) = \sum_{s \in shotclasses} P(i_s).P(j_s)$$

$P(i_s)$ being the probability that the ith frame belongs to the s shot class; and $c(i,0)$ is the cost of occlusion. The cost of occlusion is lesser if one of the frames belongs to an advertisement scene, and more otherwise. The optimal path of the match is found by backtracking the DP matrix. With this match, the observed scenes from D are warped onto the generated model M. The segments of a scene is the segment that maps to the scene in the generated model. The procedure is depicted in Figure 4. The procedure is given in Algorithm 3. In Algorithm 3, the Dynamic_Programming and Back_Track are standard dynamic programming and backtracking algorithms.

The scenes obtained from the text driven segmentation, were evaluated manually over a 20 minute video. The segments were found to be satisfactory. Out of 124 balls, the segmentation was able to identify about 98 balls correctly. The errors in identification are due to the overlap with replays and advertisements. The presence of large number of replays and advertisements, especially back to back, causes the estimation to perform poorly. It was also found that the segments generally follow the length of the

Algorithm 3. Segment_Video(V, G)

```
Set S = NULL
Identify shots, C_1, C_2, ..., C_m in V
for each shot j = 1 to m and each shot-class k = 1 to l do
    Find probability P_k(C_j), of shot C_j belonging to the k th shot-class
end for
/*Build video representation S as */
for j = 1 to L_{V_i}, and each shot-class k = 1 to l do
    Append P_k(C_j) to S_k
end for
Compute D = Dynamic_Programming(S, G)
Find optimal path using P = Back_Track(D)
/*Segment Video*/
for each scene s = 1 to n do
    Find the scene segment G_s corresponding to s in generated video G
    Find correspondence of G_s in P
    Output V_s corresponding to G_s in P
    Annotate V_s with the scene description of s
end for
```

scene model, in the absence of other discrimination. The scene model, thus, constraints the accuracy of segments obtained.

5 Automatic Annotation of Scenes

Following the text driven segmentation of video, the balls are synchronized with their commentary. This enables automatic annotations of the video scenes with their respective description. Such an annotation could be used for retrieval and summarization. It should be noted that automatic annotation of videos using visual features alone is a very difficult task using existing techniques. It is in cases like these that cross-modal techniques are highly relevant.

5.1 Retrieval

From the process of annotation, each ball has an associated textual description, which allows us to build a text based search engine over the videos. The video segments are indexed by the keywords associated with them. The keywords are obtained from the unigram frequencies of the words in the entire commentary. The most commonly occurring words are removed as stop-words. For each keyword, the associated video segments are found and indexed to it. Given a query, the index is searched for the query word, and the matched index is retrieved for the user. The user can then click on the search results and view the video segment. A screenshot of the retrieval tool is shown in Figure 5.

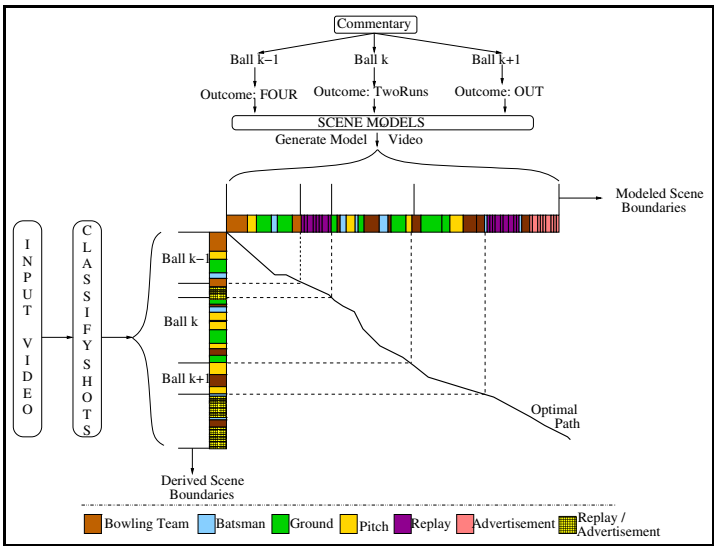


Fig. 4. The merging of visual, bottom-up and conceptual, top-down information for video segmentation using our framework

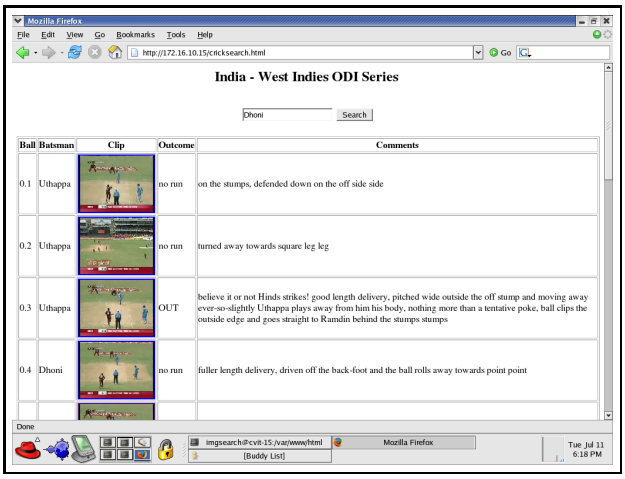


Fig. 5. Snapshot of the "Cricket Browser", the tool that allows to browse through the matches and allows for searching the annotation, thereby providing semantic access to cricket videos

The retrieval of videos using our scheme is interactive, with a retrieval time of about 0.01 seconds. This is because the search is performed in the text domain, and no image or video feature comparisons are necessary (as in general CBIR). The user can search for popular outcomes such as *four, six, out* etc., or for scenes involving his favourite

player. The user could also search in the descriptions, which means he could search for semantic concepts like an *out swinger*. Learning and identifying such subtle concepts using purely visual domain practises, is highly involved, which is circumvented by our approach. However, the accuracy of the search system is affected by the errors from the scene segmentation phase. With accurate scene segmentation, the search could provide accurate retrieval of video scenes.

5.2 Summarization

Highlights of the match are generated by finding *interesting* events from the commentary. An exciting ball is generally described in detail, with many adjectives in the sentences. These are identified using text processing schemes and the exciting balls are extracted for the highlights. We compared our highlights with those shown on TV for two matches. The evaluation measure was the number of highlights missed during the entire match. The results of the evaluations are shown in Table 1. The large disparity between the durations between the TV highlights and those generated by us is due to the fact that we have not incorporated replays into the highlights. The missed highlights are those which were not classified as *exciting* due to lack of detailed description in the commentary of some of the balls.

Table 1. Comparison of generated highlights with those created manually

Match	Input Duration	Highlights' Duration	TV Highlights' Duration	Missed Highlights
Ind Vs. WI 1	4.00	32 min	48 min	13
Ind Vs. WI 2	3.26 hr	37 min	45 min	16

6 Conclusions and Future Directions

The major contributions of our work are:

- A novel framework that performs temporal segmentation of videos into scenes by effectively merging the scene description information with visual features
- A formulation to partially bridge the semantic gap between the descriptions and the video shots they correspond to
- Automatic annotation of multimedia with text
- Search and retrieval, summarization and highlight generation of videos

One application of the results of our work is in learning semantic concepts such as a *poor shot* or a *good ball*. We have, in this work, modelled distinctive concepts for the scene models, but more robust representations are required to model concepts over shorter sequences. User preferences for highlights could be learnt using relevance feedback and customized match summaries could be generated. With an annotated corpus many video processing algorithms could be built and tested on this platform. Activity recognition systems could be reliably trained and evaluated using our corpus.

References

1. Rui, Y., Huang, T.S., Mehrotra, S.: Constructing table-of-content for videos. Multimedia Syst. **7** (1999) 359–368
2. Jiang, H., Helal, A., Elmagarmid, A.K., Joshi, A.: Scene change detection techniques for video database systems. Multimedia Syst. **6** (1998) 186–195
3. Koprinska, I., Carrato, S.: Temporal video segmentation: A survey. Signal Processing: Image Communication (2001) 477–500
4. Lefevre, S., Holler, J., Vincent, N.: A review of real-time segmentation of uncompressed video sequences for content-based search and retrieval. Real-Time Imaging **9** (2003) 73–98
5. Hanjalic, A., Lagendijk, R.L., Biemond, J.: Automated high-level movie segmentation for advanced video retrieval systems. IEEE Trans. Circuits Syst. Video Technol. **9** (1999) 580
6. Demarty, C., Beucher, S.: Morphological tools for indexing video documents. In: Proc. IEEE Intl. Conf. Multimedia Computing and Systems. (1999) 991
7. Zabih, R., Miller, J., Mai, K.: A feature-based algorithm for detecting and classifying production effects. Multimedia Syst. **7** (1999) 119–128
8. Rasheed, Z., Shah, M.: Scene detection in hollywood movies and tv shows. In: Proc. Computer Vision and Pattern Recognition. Volume 2. (June 2003) II – 343–8
9. Rui, Y., Gupta, A., Acero, A.: Automatically extracting highlights for tv baseball programs. In: ACM Multimedia, New York, NY, USA, ACM Press (2000) 105–115
10. Babaguchi, N., Kawai, Y., Kitahashi, T.: Event based indexing of broadcast sports video by intermodal collaboration. IEEE Trans. Multimedia **4** (2002) 68–75
11. Sudhir, G., Lee, J.C.M., Jain, A.K.: Automatic classification of tennis video for high-level content-based retrieval. In: Proc. International Workshop on Content-Based Access of Image and Video Databases. (1998) 81–90
12. Kolekar, M.H., Sengupta, S.: A hierarchical framework for generic sports video classification. In: ACCV (2). (2006) 633–642
13. Jadon, R.S., Chaudhury, S., Biswas, K.K.: Sports video characterization using scene dynamics. In: ICVGIP. (2004) 545–549
14. Fatemi, O., Zhang, S., Panchanathan, S.: Optical flow based model for scene cut detection. In: Canadian Conf. on Electrical and Computer Engineering. Volume 1. (1996) 470–473
15. Gunsel, B., Ferman, A., Tekalp, A.: Temporal video segmentation using unsupervised clustering and semantic object tracking. Journal of Electronic Imaging **7** (1998) 592–604
16. Lienhart, R., Kuhmunch, C., Effelsberg, W.: On the detection and recognition of television commercials. In: International Conference on Multimedia Computing and Systems. (1997) 509–516
17. Wang, L., Liu, X., Lin, S., Xu, G., Shum, H.Y.: Generic slow-motion replay detection in sports video. In: ICIP. (2004) 1585–1588
18. Li, B., Errico, J.H., Pan, H., Sezan, I.: Bridging the semantic gap in sports video retrieval and summarization. J. Vis. Commun. Image R. **15** (2004) 393–424
19. Cox, I.J., Hingorani, S.L., Rao, S.B., Maggs, B.M.: A maximum likelihood stereo algorithm. Comput. Vis. Image Underst. **63** (1996) 542–567