Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 25 (2004) 767–775
Pattern Recognition Letters
www.elsevier.com/locate/patrec

# Structure analysis of soccer video with domain knowledge and hidden Markov models

Lexing Xie [a,*], Peng Xu [a], Shih-Fu Chang [a], Ajay Divakaran [b], Huifang Sun [b]

[a] *Department of Electrical Engineering, Columbia University, New York, NY, USA*
[b] *Mitsubishi Electric Research Labs, Murray Hill, NJ, USA*

**Abstract**

In this paper, we present statistical techniques for parsing the structure of produced soccer programs. The problem is important for applications such as personalized video streaming and browsing systems, in which videos are segmented into different states and important states are selected based on user preferences. While prior work focuses on the detection of special events such as goals or corner kicks, this paper is concerned with generic structural elements of the game. We define two mutually exclusive states of the game, *play* and *break* based on the rules of soccer. Automatic detection of such generic states represents an original challenging issue due to high appearance diversities and temporal dynamics of such states in different videos. We select a salient feature set from the compressed domain, dominant color ratio and motion intensity, based on the special syntax and content characteristics of soccer videos. We then model the stochastic structures of each state of the game with a set of hidden Markov models. Finally, higher-level transitions are taken into account and dynamic programming techniques are used to obtain the maximum likelihood segmentation of the video sequence. The system achieves a promising classification accuracy of 83.5%, with light-weight computation on feature extraction and model inference, as well as a satisfactory accuracy in boundary timing.


*Keywords:* Sports video analysis; Soccer video; Hidden Markov models; Dynamic programming; Video syntax

## 1. Introduction

In this paper, we present algorithms for the analysis of video structure using domain knowledge and supervised learning of statistical models. The domain of interest here is soccer video, and the structure we are interested in is the temporal sequence of high-level game states; namely, play and break. The goal of this work is to parse the continuous video stream into a sequence of component state labels automatically, i.e., to jointly segment the video sequence into homogeneous chunks and classify each segment as one of the semantic states as well. Structure parsing is not only useful in automatic content filtering for general TV audience and soccer professionals in this special domain, it is also related to an important general problem of video structure analysis and content understanding. While most existing work

* Corresponding author.
*E-mail addresses:* xlx@ee.columbia.edu (L. Xie), pengxu@ee.columbia.edu (P. Xu), sfchang@ee.columbia.edu (S.-F. Chang), ajayd@merl.com (A. Divakaran), hsun@merl.com (H. Sun).
*URL:* http://www.ee.columbia.edu/dvmm/.

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.01.005

focuses on the detection of domain-specific events, our approach in generic high-level structure analysis is distinctive with several important advantages: (1) the generic state information can be used to filter and significantly reduce the video data. For example, typically no more than 60% of the video corresponds to play, thus we can achieve significant information reduction; (2) videos in different states clearly have different temporal variations, which can be captured by statistical temporal models such as the hidden Markov models (HMM).

Related work in the literature of sports video analysis has addressed soccer and various sports games. For soccer video, prior work has been on shot classification (Gong et al., 1995), scene reconstruction (Yow et al., 1995), and rule-based semantic classification (Qian and Tovinkere, 2001). For other sports video, supervised learning was used by Zhong and Chang (2001) to recognize canonical views such as baseball pitching and tennis serve. In the area of video genre segmentation and classification, Wang et al. (2000) have developed HMM-based models for classifying videos into news, commercial, sports and weather reports.

In this work, we first exploit domain-specific video syntax to identify salient high-level structures. Such syntactic structures are usually associated with important semantic meanings in specific domains. Taking soccer as a test case, we identify play and break as two recurrent high-level structures, which correspond well to the semantic states of the game. Such observations then lead us to choosing two simple, but effective features in the compressed domain, dominant color ratio and motion intensity. In our prior work (Xu et al., 2001), we showed such specific set of features, when combined with rule-based detection techniques, were indeed effective in play/break detection in soccer. In this paper, we will use formal statistical techniques to model domain-specific syntactic constraints rather than using heuristic rules only. The stochastic structure within a play or a break is modelled with a set of HMMs, and the transition among these HMMs is addressed with dynamic programming. Average classification accuracy per segment is 83.5%, and most of the play/break boundaries are correctly detected within a 3-second window (Xie et al., 2002). It is encouraging that high-level domain-dependent video structures can be computed with high accuracy using compressed-domain features and generic statistical tools. We believe that the performance can be attributed to the match of features to the domain syntax and the power of the statistical tools in capturing the perceptual variations and temporal dynamics of the video.

In Section 2, we define the high-level structures of play and break in soccer, and present relevant observations of soccer video syntax; in Section 3 we describe algorithms for feature extraction and validation results of such a feature set with rule-based detection; in Section 4 we discuss algorithms for training HMMs and using the models to segment new videos to play and break; experiments and results are presented in Section 5; and in Section 6 we draw conclusions and discuss future work.

## 2. The syntax and high-level structures in soccer video

In this section, we present a few observations on soccer video that explore the interesting relations between syntactic structures and semantic states of the video.

### 2.1. Soccer game semantics

We define the set of mutually exclusive and complete semantic states in a soccer game: play and break (FIFA, 2002). The game is in play when the ball is in the field and the game is going on; break, or out of play, is the compliment set, i.e., whenever "the ball has completely crossed the goal line or touch line, whether on the ground or in the air" or "the game has been halted by the referee".

Segmenting a soccer video into play/break is hard because of (1) the absence of a canonical scene as opposed to the serve scene in tennis (Sudhir et al., 1998) or the pitch scene in baseball video (Zhong and Chang, 2001); (2) the loose temporal structure, i.e. play/break transitions and highlights of a game (goal, corner kick, shot, etc.)

do not have a deterministic relationship with other events. This is different from the case for tennis—volleys are always preceded by a serve in a tennis game. Yet identifying play/break is interesting because not only can we achieve about 40% information reduction (Table 1), play/break information also has many interesting applications such as play-by-play browsing and editing, or analysis of game statistics.

### 2.2. Soccer video syntax

Soccer video syntax refers to the typical production style and editing patterns that help the viewer understand and appreciate the game. Two major factors influencing the syntax are the producer and the game itself, and the purpose of syntax is to emphasize the events as well as to attract viewers' attention (such as the use of cutaways). Specifically, soccer video syntax can be characterized by some rules-of-thumb observed by sports video producers (Shook, 1995): (1) convey global status of the game; (2) closely follow action and capture highlights. In our algorithm, two salient features are selected to capture this syntax implicitly.

One additional informative observation is about the three main types of views under the common camera setup in soccer video production; namely *global* (long shot), *zoom-in* (medium shot), and *close-up*, according to the scale of the shot. Distinguishing the three types of views is helpful because of the following reasons:

(1) The type of view is closely related to the semantic state of the game. During the play, TV producers tend to stay in the global view in order to keep the audience informed of the status of the entire field; interrupted by short *zoom-ins* or *close-ups* to follow the players' action; during the break, there are less global views because not much is happening in a global scale, and zoom-ins and close-ups tends to be the majority as they can effectively show the cause and effect of the break (such as why a foul would happen, its consequences, etc.); furthermore, the transitions between plays and breaks usually arise within some time range, if not perfectly aligned, with a transition of view type.

(2) The type of view can be approximately computed with low-level features. The difference among the views is usually reflected in the ratio of green grass area in soccer video, as shown in Fig. 1. In Section 3.1, the algorithms for computing grass-area ratio and statistics on how this feature relates to the scale of the view will be presented in more detail. We shall keep in mind, however, the scale of a shot is a semantic concept. This is partly because the long-medium-close-up scales are de-

Table 1
Soccer video clips used in the experiment

| Clip name | Length | # of plays | Play-percentage (%) | Source | Dominant hue |
|---|---|---|---|---|---|
| Argentina | 23′56″ | 34 | 58.5 | TV program | 0.1840 |
| Korea A | 25′00″ | 37 | 60.6 | MPEG-7 | 0.2436 |
| Korea B | 25′23″ | 28 | 52.1 | MPEG-7 | 0.2328 |
| Espana | 15′00″ | 16 | 59.2 | MPEG-7 | 0.2042 |

Fig. 1. Dominant color ratio as an effective feature in distinguishing three kinds of views in soccer video. Left to right: global, zoom-in, close-up. Global view has the largest grass area, zoom-in has less, and close-ups has hardly any (including cutaways and other shots irrelevant to the game).

fined with respect to the size of the subject. Different viewers or even the same viewer under a different context may disagree on what the subject and the appropriate distances that characterize the scale of the shot shall be. "The dividing line between *long shot* and *medium shot* is blurry, as is the line between *medium shot* and *close shot*... There is no evident reason for this variation. It is not a distinction, for example, between TV and film language or 1930s and 1980s language." (The Wikipedians, 2002). Furthermore, the mapping from *views* to features adds one more factor of uncertainty to this inherent flexibility in the definition of *views*, thus trying to map features to views will be a pattern recognition problem in itself.

## 3. Computing informative features

Based on observations relating soccer video semantics, video production syntax and low-level perceptual features, we use one special feature, *dominant color ratio*, along with one generic feature, *motion intensity*, to capture the characteristics of soccer video content. Moreover, out attention here is on compressed-domain features, since one of the objectives of the system is real-time performance under constrained resource and diverse device settings.

### 3.1. Dominant color ratio

Computing dominant color ratio involves two steps, i.e. learning the dominant color for each clip, and then use the learned definition for each clip to find the percentage of pixels of this color.

#### 3.1.1. Adaptively learning dominant color

The grass color of the soccer field is the dominant color in this domain, since a televised soccer game is bound to show the soccer field most of the time, in order to correctly convey the game status. The appearance of the grass color, however, ranges from dark green to yellowish green or olive, depending on the field condition and capturing device. Despite these factors, we have observed that within one game, the hue value in the HSV (Hue-Saturation-Value) color space is relatively stable despite lighting variations; hence, learning the hue value would yield a good definition of dominant color.

The dominant color is adaptively learned for each video clip, using the following cumulative color histogram statistic: 50 frames are drawn from the initial five minutes (an I/P frame pool of 3000) of the video, the 128-bin hue histogram is accumulated over all sample frames, and the peak of this cumulative hue histogram correspond to the dominant color. This experiment is repeated eight times, each with a different set of frame samples; two standard deviations below and above the mean of the peak hue value over the eight trials is taken as the range for grass color in the current video clip; this definition will include 95.4% of the grass pixels, assuming the distribution of peak hue value is Gaussian. This definition of dominant color is specific enough to characterize variations across different games, yet relatively consistent to account for the small variations within one game (Table 1). We have also performed this test for two soccer videos that comes from the same game, 30 min apart, and results indeed show that the difference of the mean hue values over time is smaller than the standard deviation within one clip.

#### 3.1.2. The dominant color ratio

Once we can distinguish grass pixels vs. non-grass pixels in each frame, the feature dominant-color-ratio $\eta$ is just computed as $\eta = |P_g|/|P|$, where $|P|$ is the number of all pixels, and $|P_g|$ is the number of grass pixels.

#### 3.1.3. The effectiveness of dominant color ratio

Observations in Section 2.2 showed intuitions that relates $\eta$ to the scale of view and in turn to the status of the game. Experiments in (Xu et al., 2001) showed accuracies of 80–92% in labelling the three kinds of views using adaptive thresholds, and accuracies 67.3–86.5% in segmenting play/breaks from the view labels using heuristic rules. While the results are satisfactory, it is worth noticing that (1) the scale of view is a semantic concept (Section 2.2), and most of the errors in labelling views is due to model breakdown; for example, *zoom-in* is sometimes shot with a grass background, and the

area of the grass is sometimes even larger than that of the global shots; (2) it is not sufficient to model temporal relationships between views and game state with heuristic rules, as most of the errors is caused by violation of the assumption that a play-break transition only happens upon the transition of views. On the other hand, shots and shot boundaries have similar drawbacks such as (1) shot boundaries are neither aligned with the transitions of play/break nor with switches in the scale of view; (2) shot detectors tend to give lots of false alarms in this domain due to unpredictable camera motion and intense object motion. Hence, in our algorithms detailed in Section 4, each game state corresponds to a feature value distribution of a mixture of Gaussians, while the temporal transitions are modelled as a Markov chain.

Note the dominant color feature can be generalized to many other types of sports as a good indicator of game status such as baseball, American football, tennis, basketball, etc.

### 3.2. Motion intensity

Motion intensity $m$ is computed as the average magnitude of the *effective* motion vectors in a frame:

$$m = \frac{1}{|\Phi|} \sum_{\Phi} \sqrt{v_x^2 + v_y^2},$$

where $\Phi = \{\text{inter-coded macro-blocks}\}$ and $v = (v_x, v_y)$ is the motion vector for each macro-block.

This measure of motion intensity gives an estimate of the gross motion in the whole frame, including object and camera motion. Moreover, motion intensity carries complementary information to the color feature, and it often indicates the semantics within a particular shot. For instance, a wide shot with high motion intensity often results from player motion and camera pan during a play; while a static wide shot usually occurs when the game has come to a pause. In the sample clip shown in Fig. 3, we can see distinct feature patterns are associated with the scale of shot and the game status. But as these variations are hard to quantify with explicit low-level decision rules, we resort to HMM modelling described in the next section. Here we directly estimate motion intensity from the compressed bitstream instead of the pixel domain, since MPEG motion vectors can approximate sparse motion field quite well, and accurate motion estimation in the pixel domain is usually orders of magnitude more complex, depending on the choice of algorithm.

## 4. Play-break segmentation with HMMs

In a sense, distinguishing the distinct inherent states of a soccer game, play ($P$) and break ($B$), is analogous to isolated word recognition in (Rabiner, 1989). Here each model corresponds to a class—phoneme in the speech case, $P$ or $B$ in a soccer video; the sub-structures within each model accounts for transitions and variations within and between phonemes in speech, and the switching of shots and the variations of motion in a soccer game. This analogy leads to our use of HMMs for soccer video segmentation, yet our case has one more uncertainty factor: there is no pre-segmented *word-boundary* in soccer.

Fig. 2 is an overview of the algorithm that takes the continuous feature stream, and segments and classifies it into play/break segments. Where the left half evaluates the data likelihood of fixed-length short segments against pre-trained HMMs (Section 4.1); and the right half makes use of long-term correlation to smooth labels for individual segments, and generate final segmentation (Section 4.2).

### 4.1. HMM

The HMMs are trained with manually labelled data set using the Expectation–Maximization (EM) algorithm. Since the domain-specific classes P/B in soccer are very diverse in themselves (typically ranging from 6 s up to 2 min in length), we use a set of models for each class to capture the structure variations, instead of just using a homogeneous model for each class as in (Wang et al., 2000). We trained $K = 6$ HMMs with different topologies for play and for break, respectively. These include: 1/2/3-state fully connected models, 2/3-state left-right models and a 2-state

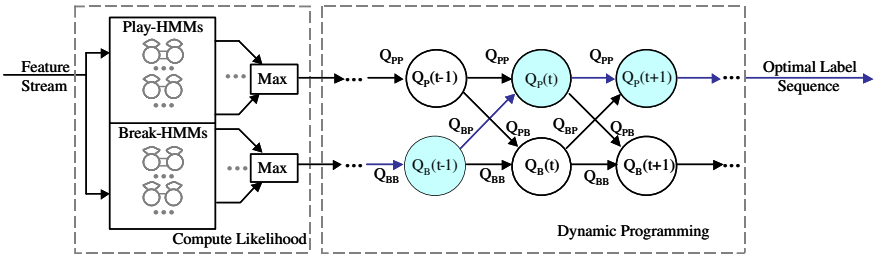


Fig. 2. Segmentation using HMM-dynamic programming. Only 2 out of 6 HMM topologies are shown; the Qs are HMM model likelihoods or transition likelihoods.

fully connected model with an entering and an exiting state. The observations are modelled as a mixture of Gaussians, and we use two mixtures per state in the experiments. Training data are manually chopped into homogeneous play/break chunks; model parameters are trained using the iterative EM algorithm over the pool of labelled data.

Once HMM models are learned, they can be used to parse new video clips into play/break segments. Prior to entering the classification stage, each feature dimension is smoothed with a low-pass filter for denoising and normalized with respect to its mean and variance to account for inter-clip variations. And then, we window the feature sequence with a sliding window of length $N$, sliding by $M$ samples, resulting in a set of short feature chunks $\{F(t_w)\}_{t_w=1}^{T_w}$, each of size $D \times N$, where $D$ is the feature dimension, $T$ is the total number of samples in the whole video sequence, $t_w$ is the index of sliding windows, and $T_w = \lceil T/M \rceil$ is the total number of windowed feature vectors.

We evaluate the data likelihood for each of the set of pre-trained *play* models $\Theta_P = \{\Theta_P^1, \ldots, \Theta_P^K\}$ against each feature chunk $F(t)$, to get likelihood values $\overline{Q}_P^k(t)$, $k = 1, \ldots, K$; $t = 1, \ldots, T_w$. Similarly, for *break* models $\Theta_B = \{\Theta_B^1, \ldots, \Theta_B^K\}$, the likelihood values are $\overline{Q}_B^k(t)$, $k = 1, \ldots, K$; $t = 1, \ldots, T_w$. We could have taken the maximum-likelihood decision among all HMMs as the label for the current feature chunk, but this simple strategy would lose the correlation present beyond $N$ samples. We therefore keep the best likelihood value in each of the P/B class as the *best fit among mixture of experts*, i.e. $Q_P(t) \triangleq \max_k \{Q_P^k(t)\}$, $Q_B(t) \triangleq \max_k \{Q_B^k(t)\}$, $t = 1, \ldots, T_w$ and model longer term correlation on top of them with dynamic programming as presented in Section 4.2.

There are subtle choices in the HMM training-classification process: (1) Training is not conducted over $N$-sample windows since we hope that the HMM structures can take longer time correlation into account, and thus "tolerate" some less frequent events in a semantic state such as short close-ups within a play. Experiments show that the overall accuracy will be consistently 2–3% lower if models are trained on short segments, and the video tends to be severely over-segmented as some of the short close-ups and cutaways during a play will be misclassified as break. In our separate experiments, a *Student's t*-test shows that the null hypothesis that training on longer and short segments have the same accuracy is rejected with 95.0% confidence. (2) Since training is done for the whole play or break, but classification is done over short segments, we may conjecture that results will not be worse if only the three fully connected models (instead of all six) are used. This is confirmed by the result that classification accuracy only differs by 1.5% for these two cases, but the significance for such as test *cannot* be established since the $p$-value for $t$-test is less than 50%.

### 4.2. Find optimal path with dynamic programming

HMM likelihood represents the *fitness* of each model for every short segment, but the long-term correlation is unaccounted for. Thus, the remaining problem is to find a global optimal state path $\{s(t)\}_{t=1}^{T_w}$ using neighborhood information.

If we assume the transition between states P and B across windowed segments has Markov prop-

erty, then this problem can be solved with well-known dynamic programming techniques (Rabiner, 1989), as illustrated in Fig. 2. Here we only need to keep track of the *cumulative node score* $\sigma_s(t)$, the *best* score for any path ending in state $s$ at $t$, and *back-pointers* $\delta_s(t)$, the previous node on this *best* path, for $s \in \{\mathrm{P}, \mathrm{B}\}$, $t = 1, \ldots, T_\mathrm{w}$. The quantities $\sigma_s(t)$ and $\delta_s(t)$ are computed iteratively for each $t$ on a $2 \times T_\mathrm{w}$ trellis, and the optimal path $s(t)$ is obtained by backtracking from the final node.

Note we use the *best fit among mixture of experts* $Q_\mathrm{P}(t)$, $Q_\mathrm{B}(t)$ as node likelihoods at time $t$, and transition probabilities $Q_{\mathrm{PP}}$, $Q_{\mathrm{PB}}$, $Q_{\mathrm{BP}}$, $Q_{\mathrm{BB}}$ are obtained by counting over the training set:

$$Q_{s',s} \triangleq \log P(s(t+1) = S | s(t) = S') = \log\left(\sum_{t=1}^{T_\mathrm{w}-1} \frac{I_{s(t+1)=S} \cdot I_{s(t)=S'}}{I_{s(t)=S'}}\right), \quad (1)$$

where $S, S' \in \{\mathrm{P}, \mathrm{B}\}$, and indicator function $I_c = 1$ when the statement $c$ is true, 0 otherwise.

The iteration at time $t$ is done as in Eqs. (2) and (3), i.e., for each state in the state space at time $t$, we keep track of the incoming path with the maximum likelihood (Eq. (3)), and the corresponding likelihood is also recorded as the *score* for the current node.

$$\sigma_s(t) = \max_{s' \in \{\mathrm{P},\mathrm{B}\}} \left\{\lambda \cdot Q_{s',s} + (1-\lambda) \cdot Q_s(t) + \sigma_{s'}(t-1)\right\}, \quad (2)$$

$$\delta_s(t) = \arg\max_{s' \in \{\mathrm{P},\mathrm{B}\}} \left\{\lambda \cdot Q_{s',s} + \sigma_{s'}(t-1)\right\}. \quad (3)$$

Here the transitions are only modelled between play and break, rather than among all of the underlying HMM models, because having this $2\times2$ transition matrix is sufficient for our play/break segmentation task, and modelling all possible transitions among all HMMs (a $12\times12$ transition matrix required) is subject to over-fitting. Intuitively, if the scores $Q_\mathrm{P}(t)$ and $Q_\mathrm{B}(t)$ at each node were the true posterior probability that feature vector at time $t$ comes from a play or a break model, then this dynamic programming step would essentially be a second-level HMM. Moreover, the constant $\lambda$ weights model likelihood and transition likelihood: $\lambda = 0$ is equivalent to maximum likelihood classification; $\lambda = 1$ gives a first-order Markov model. Classification accuracy is not very sensitive to $\lambda$ if its value is within a reasonable range. A typical $\lambda$ is 0.25, and classification accuracy only varies within 1.5% for $\lambda \in [0.1, 0.4]$.

As shown in Fig. 3, employing this dynamic programming step alleviates over-segmentation, and the results show that average classification accuracy is improved by 2.2% over HMM-maximum likelihood only, with a $t$-test confidence 99.5%. As shown in Fig. 2, this dynamic programming step involves going through all $Q$ predecessors for each of the $Q$ states for each of the $T_\mathrm{w}$ windowed feature vectors. Hence its complexity is $O(T_\mathrm{w} \cdot Q^2)$, linear in $T_\mathrm{w}$ when $Q$ is a small constant ($Q = 2$ in our case). Furthermore, the computation load of dynamic programming is only a fraction of the HMM classification step, since the latter will be $O(T \cdot C)$, where $T \approx M \cdot T_\mathrm{w}$, and the multiplying constant $C$ is quadratic to the maximum number of states used in the HMMs.

In addition, we have also looked into the problem of finding maximum-likelihood segmentation from an unevenly spaced grid. We use

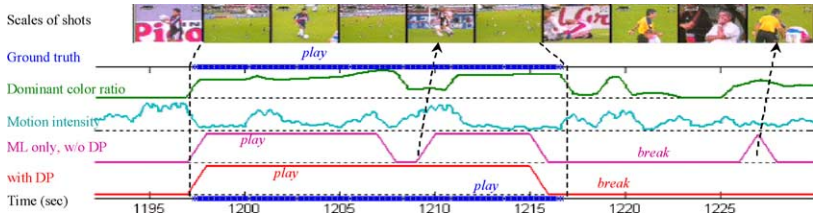


Fig. 3. Part of clip Argentina ($19'52'' \sim 20'30''$): key frames, feature contours, and segmentation results (with or without dynamic programming).

shot boundary detection results as hypothesis transition points (a hypothesis set much sparser than every $N$-sample segment in our experiments), and search through not only all *previous states* but also *all durations* with an algorithm very similar to the inhomogeneous HMM (Ramesh and Wilpon, 1992). This incurs a computational complexity of $O(T_w^2)$ instead of $O(T_w)$, with significantly more running time on our current data set. The accuracy, however, was 2% less than that on a homogeneous grid. This deterioration is partly due to the mismatch between shot boundaries and play/break boundaries, but increasing the number of hypothesis transition points is not worthwhile due to the increased computation load.

## 5. Experiments

Four soccer video clips used in our experiment are briefly described in Table 1. All clips are in MPEG-1 format, SIF size, 30 frames per second or 25 frames per second. The dominant hue values are adaptively learned for each clip (Section 3.1) and the dominant color ratios are computed on $I$- and $P$-frames only. The motion intensities are computed on $P$-frames and interpolated on $I$-frames. A window of three seconds long sliding by one second is used to convert continuous feature stream into short segments. The ground-truth is labelled under the principles that (1) we assume the game status does not change unless indicated by a perceivable event, e.g., the ball is shown out of boundary, a whistle sound is heard, etc.; (2) replays are treated as in play, unless it is not adjacent to a play and shorter than 5 s. Here play-percentage refers to the amount of time the game is in play over the total length of the clip.

In our experiments, the HMM are trained on *one entire clip* and tested on all three other clips; this process is repeated for each video as the training set. We measure *classification accuracy* as the number of correctly classified samples over total number of samples. Training and testing accuracies are shown in Table 2. Average classification performance (*avg-cla*) of each clip as test set is computed as the mean of the non-diagonal elements of the current row; similarly, average generalization performance (*avg-gen*) is computed for the clip as training set; and the overall average classification/generalization accuracy over the entire data set is put in the lower right corner.

Table 2
Classification accuracy, the *diagonal element*s are training results

| Test set | Training set | | | | avg-cla |
|---|---|---|---|---|---|
| | Argentina | Korea A | Korea B | Espana | |
| Argentina | *0.872* | 0.825 | 0.825 | 0.806 | 0.819 |
| Korea A | 0.781 | *0.843* | 0.843 | 0.798 | 0.807 |
| Korea B | 0.799 | 0.853 | *0.853* | 0.896 | 0.849 |
| Espana | 0.799 | 0.896 | 0.896 | *0.817* | 0.863 |
| avg-gen | 0.793 | 0.858 | 0.855 | 0.833 | 0.835 |

Since our goal is to do joint segmentation and classification in one-pass, we are also interested in measuring the boundary accuracy. For each 3-s segment (1 s apart from each other), the classifier not only gives the P/B label, but also indicates if a boundary exists between the previous and the current label. This is different from boundary detection algorithms that solely aim at outlier detection (such as shot boundary detection by measuring histogram distance), since each misjudgment here can cause two false positives instead of one. Let boundary-offset be the absolute difference between the nearest boundary in detection result and every boundary in the ground truth. The distribution over all testing trials is shown in Table 3.

The results show that our classification scheme has consistent performance over various dataset; models trained on one clip generalize well to other clips. The classification accuracy is above 80% for every clip, and more than 60% of the boundaries are detected within a 3-s ambiguity window (Table 3). Compared to the heuristic rules described in Section 3.1, testing accuracy improves 1%, 15%, and 18% for clips Korea B, Argentina and Espana (trained on Korea A), respectively. Typical errors in the current algorithm are due to model breakdowns that feature values do not al-

Table 3
Boundary offset distribution

| Offset (s) | [0,3] | (3,6] | (6,10] | (10,25] | (25,50] | >50 |
|---|---|---|---|---|---|---|
| Percentage (%) | 62 | 12 | 5.8 | 13 | 6.7 | 0.7 |

ways reflect semantic state of the game such as a brief switch of play/break without significant change in features.

## 6. Conclusion

In this paper, we presented new algorithms for soccer video segmentation and classification. First, play and break are defined as the basic semantic elements of a soccer video; second, observations of soccer video syntax are described and feature set is chosen based on these observations; and then, classification/segmentation is performed with HMM followed by dynamic programming. The results are evaluated in terms of classification accuracy and segmentation accuracy; extensive statistical analysis show that classification accuracy is about 83.5% over diverse data sets, and most of the boundaries are detected within a 3-s ambiguity window. This result shows that high-level video structures can be computed with high accuracy using compressed-domain features and generic statistical tools, domain knowledge plays the role of matching features to the domain syntax and selecting the statistical models in capturing the visual variations and temporal dynamics of the video.

The algorithms leaves much room for improvement and extension: (1) There are other relevant low-level features that might provide complementary information and may help improve performance, such as camera motion, edge, or audio features; (2) Higher-level object detectors, such as goal and whistle detection, can be integrated; (3) It will be worthwhile to investigate unsupervised learning scenarios without extensive training.

## References

FIFA, 2002. Laws of the game. Federation Internationale de Football Association, http://www.fifa.com/fifa/handbook/laws/2002/LOTG2002_E.pdf.

Gong, Y., Lim, T., Chua, H., May 1995. Automatic parsing of TV soccer programs. In: IEEE International Conference on Multimedia Computing and Systems, pp. 167–174.

Qian, R.J., Tovinkere, V., August 2001. Detecting semantic events in soccer games: Towards a complete solution. In: IEEE International Conference on Multimedia and Expo.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77 (2), 257–285.

Ramesh, P., Wilpon, J., 1992. Modeling state durations in hidden Markov models for automatic speech recognition. In: ICASSP, Vol. I, pp. 381–384.

Shook, F. (Ed.), 1995. Television field production and reporting, 2nd Edition. Longman Publisher USA, Sports Photography and Reporting, Chapter 12.

Sudhir, G., Lee, J.C.M., Jain, A.K., January 1998. Automatic classification of tennis video for high-level content-based retrieval. In: IEEE International Workshop on Content-Based Access of Image and Video Database.

The Wikipedians, 2002. Wikipedia, the free encyclopedia. Film section, http://www.wikipedia.org/wiki/Medium_shot/.

Wang, Y., Liu, Z., Huang, J., 2000. Multimedia content analysis using both audio and visual clues. IEEE Signal Processing Magazine 17 (6), 12–36.

Xie, L., Chang, S.-F., Divakaran, A., Sun, H., 2002. Structure analysis of soccer video with hidden Markov models. In: Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP). Orlando, FL.

Xu, P., Xie, L., Chang, S.-F., Divakaran, A., Vetro, A., Sun, H., 2001. Algorithms and systems for segmentation and structure analysis in soccer video. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME). Tokyo, Japan.

Yow, D., Yeo, B.-L., Yeung, M., Liu, B., 1995. Analysis and presentation of soccer highlights from digital video. In: Asian Conference on Computer Vision.

Zhong, D., Chang, S.-F., August 2001. Structure analysis of sports video using domain models. In: IEEE International Conference on Multimedia and Expo.