論文 / 著書情報
Article / Book Information

Title(English)	Robust Scene Recognition Using Scene Context Information for Video Contents
Authors(English)	Koichi Shinoda, Ryoichi Ando, Sadaoki Furui, Takahiro Mochizuki
Citation(English)	Proc. International Symposium on Large-Scale Knowledge Resources(LKR2007), Vol. , No. , pp. 107-112
発行日 / Pub. date	2007, 3

Robust Scene Recognition Using Scene Context Information for Video Contents

Koichi Shinoda, Ryoichi Ando, Sadaoki Furui
Department of Computer Science,
Tokyo Institute of Technology

Takahiro Mochizuki
NHK Science & Technical
Research Laboratories

Abstract

We propose a robust scene recognition framework using scene context information for multimedia contents. In multimedia contents, some scene sequences are more likely to happen compared with other scene sequences. We employ a statistical approach to deal with this scene context information. We employ a hidden Markov model (HMM) to model each scene and an *n*-gram language model to represent the scene context. We evaluated the proposed method in scene recognition experiments for 16 scenes in video data of 25 baseball games. The proposed method significantly improved the results compared to that without scene context information.

Index Terms: CBVIR, sports video, indexing, HMM, *n*-gram model

1. Introduction

Recent advances in computer technology, particularly in storage technology, have resulted in significant increase in the number and quality of video databases. As it is difficult for ordinary people to browse the entire content of each video database, database indexing is strongly required for searching and summarization. The construction of such indexes is mostly carried out by human experts who manually assign a limited number of keywords to the video content, and it is an expensive and time consuming task. Therefore, automatic indexing using pattern recognition techniques for video content, which is called content-based video information retrieval (CBVIR), has been studied extensively [1]. In this paper, we focus on scene recognition from scene sequences of video data.

It is well known that context information is effective in pattern recognition. For detecting and classifying regions and objects in a static image, Kumar *et al.* [2] proposed a robust object detection method, where three different types of contextual interactions, *region-region*, *object-region* and *object-object* are used. Since video data is a sequence of static images, additional context information that represents relations between video frames at different times is also expected to be effective for CBVIR. For example, suppose there are three successive scenes; a scene of an airplane gliding, a scene of the airplane taking off and a scene in which the airplane is obscured by clouds. The last scene should be recognized as a scene of an airplane flying when the recognition result of the previous two scenes are available even though it is difficult to identify the airplane in the image.

Multimedia content is essentially a communication tool between human beings. Thus, we believe that most of it has *messages* that should be conveyed from one person to another. Assuming that a video has common *language* and common *grammar* according to video types, the statistical approaches used in speech recognition or natural language processing is expected to be applicable for modeling scene content.

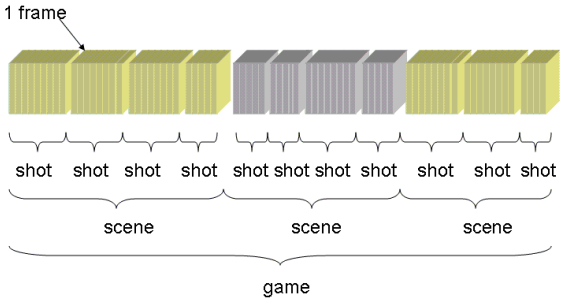

Figure 1: Structure of baseball broadcast video.

In our previous work, we studied scene recognition for a video broadcast of a baseball game [3]. We employed multi-stream HMMs to model each scene using global features and dynamic features as input. This method was evaluated using digest data that contained only highlight scenes of the game. However an entire baseball game includes replays, CG-effects, strikes, balls, and fouls, so it is necessary to achieve robustness against various types of scenes. In this paper, we propose a CBVIR method using scene context information for baseball broadcast video, which is deeply related to the statistical framework mainly used in speech recognition. We employ a statistical language model constructed from a large amount of training data for robust modeling of scene contexts. Since the structure of a baseball broadcast is relatively simple, scene contexts can be clearly defined; and thus, the proposed method is expected to be effective.

This paper is organized as follows. Section 2 presents related work on CBVIR for sports content. Section 3 explains our framework. Section 4 explains the language model we employed. Section 5 reports our experimental results. Finally, Section 6 summarizes our work.

2. Related works

Recently, CBVIR for sports video has been extensively studied. Its targets include baseball [3–7], football [8, 9], tennis [10], basketball [11], and American football [12]. In this paper, our target for CBVIR is baseball broadcast video. In a baseball broadcast, the minimum unit is a *frame*, a static image. Multiple frames recorded by a single fixed camera form a *shot*. A sequence of these shots forms a *scene* (Fig. 1). A scene consists of shot sequences between a pitching shot and a next pitching shot. The contexts or transitions between those shots provide useful information for scene recog-

Figure 2: Examples of camera shots of a *home run* scene, a *walk* scene, and an *ground out* scene.

nition (Fig. 2). For baseball scene recognition, Chang *et al.* [5] proposed an HMM-based method. In their study, first, video data is segmented into shots. Then, recognition is applied to these shot sequences based on HMMs in which each state represents a *shot type*. Xu *et al.* [8] also proposed an HMM-based method to distinguish *play* and *non-play* scenes for a football broadcast video . In these works, domain-specific knowledge about shot types and transitions among them were used intensively to improve the system's performance. Gong *et al.* [9] proposed an automatic detection and classification method using image, audio, and speech features based on a maximum entropy model (MEM). Few CBVIR methods have used scene context information. Liang *et al.* [7] proposed a rule-based method using superimposed captions, in which only three types of information were considered: the number of outs, the number of scores, and the situation of runners on base. Kijak *et al.* [10] proposed a scene recognition method based on HMMs for a tennis broadcast video in which broadcast video structures and rules of tennis were used to connect HMMs hierarchically.

3. Robust Scene Recognition

3.1. Framework

Inspired by the success of applying statistical frameworks in the speech recognition field [13], we proposed the following *data-driven* approach to provide a *robust* scene recognition system [3]. In this approach, we regarded a shot as being analogous to a *phone*, and a scene to a *word*. Based on this assumption we utilized the framework of *continuous speech recognition* in scene recognition. Given a sequence of observed feature vectors $O = o_1, \ldots, o_m$ (m is the number of frames), the probability of scene sequences $H = h_1, \ldots, h_n$ is

$$P(H|O) \propto P(O|H)P(H), \tag{1}$$

where $P(O|H)$ is the probability of O being observed in scene sequence H, and $P(H)$ indicates the probability of sequence H. $P(O|H)$ is computed by a *video model*, whereas $P(H)$ is computed by a *language model* that represents the scene context. The sequence H that maximizes $P(H|O)$ is the recognition result. In our previous work, a simple grammar represented by the Chomsky expression $< H_1|H_2|...|H_n >$ was used as a language model. This grammar indicates that the scene sequence in data is a combination of many scenes in an arbitrary order. Therefore, this grammar cannot represent scene contexts. To represent scene contexts, there are three typical language models: the n-gram model, HMM, and probabilistic context-free grammar. Here, we employ n-grams as a widely used language model.

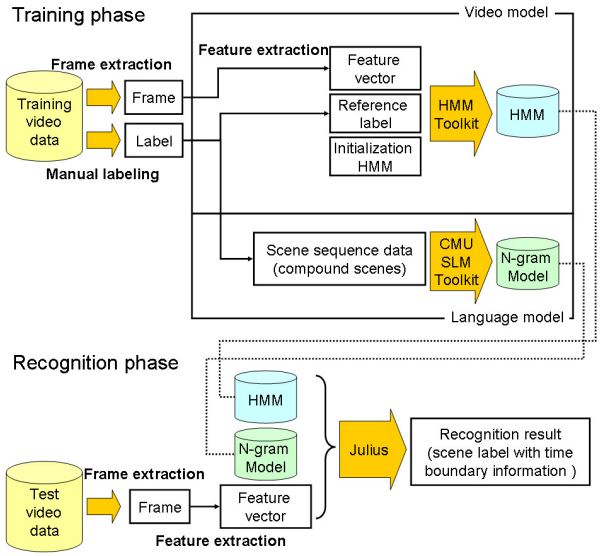

Figure 3: System overview

3.2. System overview

Our scene recognition system consists of two phases (Fig. 3):

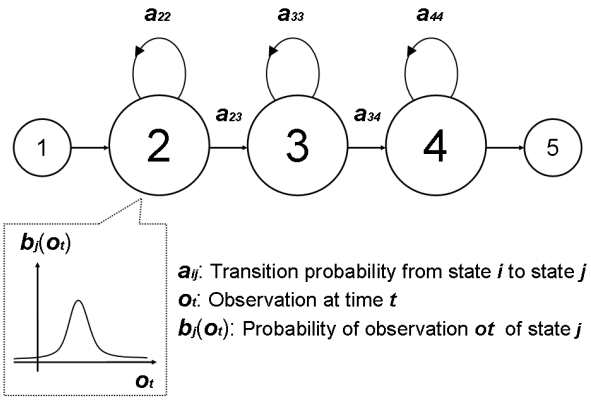

Figure 4: A left-to-right 3-state HMM having a single Gaussian output probability.

1. In the training phase, we construct video models and language models. For video models, frames are extracted from video data, and a feature vector is extracted for each frame. One HMM is prepared for each scene, and its parameters are estimated on the basis of a training set of feature vectors and on reference labels with boundary information that has been prepared manually. For language models, *n*-gram models are also constructed by scene sequences in these reference labels. The appearance probability of scenes that occur after a certain scene sequence is calculated to represent the scene context.

2. In the recognition phase, we extract frames from test video data and calculate feature vectors in exactly the same way as in the training phase. Then, using the trained HMMs and trained *n*-gram models, we conduct scene recognition on the test data. Given a feature vector sequence, the corresponding scene sequence is recognized by the *Viterbi algorithm*. The output is a sequence of scene labels with time boundary information.

3.3. Video model

In this study, we employed HMMs to model scenes [3]. HMMs are effective models for time-varying patterns and have been used widely to model scenes of sports video [4, 5, 10, 14]. In conventional HMM-based scene recognition methods (e.g., [5]), each state of an HMM is usually assigned to a specific shot type, and the HMM of each scene label has a specific topology that is determined heuristically. Inspired by the effectiveness of the data-driven approach used in speech recognition, we do not explicitly define a specific topology for each scene label, but use a common left-to-right HMM for all scene labels (Fig. 4). The reason for this is that, in real data, while the shot transition of each scene varies greatly, few clues about the underlying shot transition are apparent. Using this data-driven approach makes it easy to prepare scene models and to achieve robustness against unknown data. Our framework can be applied without any modification to recognize new scene labels when the amount of available training data increases.

In the HMM, each state j has an associated observation probability distribution $b_j(\mathbf{o}_t)$ that determines the probability of generating observation $\mathbf{o}_t$ at time t, and each pair of states i and j has an associated transition probability a_{ij}. For each scene model H, the HMM parameters a_{ij} and b_j are estimated from training data using the *Baum-Welch algorithm*. A single Gaussian distribution is used as the output probability.

3.4. Low level features

To make our framework generally applicable, we avoid using any game-specific features, such as those related to infield color or uniform color. We use only global features, and we do not use features related to specific objects because it is not always easy to extract these objects from video images under various conditions.

3.4.1. Low frequency features (LFs)

Low frequency components are extracted as global features from an image of each video frame using discrete cosine transform (DCT). First, to reduce computational costs, the image is compressed from 720×480 pixels to one tenth of that size, 72 × 48 pixels. Then, luminance is extracted from an RGB image to create a gray-scale image. The gray-scale image is transformed into a frequency domain using DCT, and 30-dimension low frequency components are used as features.

3.4.2. Dynamic low frequency features (DLFs)

While LFs are expected to be sufficient for representing global information for a still image, such representation for video images requires additional information for describing objects moving in a video stream. First, successive two-frame images are compressed from 720×480 pixels to one tenth of that size. Then, luminance is extracted from an RGB difference image between two successive images to create a gray-scale image. The gray-scale image is transformed into a frequency domain using DCT, and 30-dimension low frequency components are used as dynamic features.

3.4.3. Camera motion features (CFs)

Camera motions in a baseball broadcast consist of three types of motion: pan, tilt, and zoom. Camera shots of the same scene label tend to have similar camera motions. During a pitching shot, for example, the camera usually does not move. In a shot in which a batter runs from home base to first base, the camera pans from left-to-right. When a batter hits a fly ball, the camera tilts upward. Camera motion information has been proven to be effective information for shot segmentation and categorization in sports video [5, 15, 16]. We expect that this feature will also be effective in our scene recognition. To represent camera motions, we used optical flows calculated using the Lucas-Kanade method [17]. First, successive two-frame images are compressed from 720×480 pixels to one third of that size, 240 × 160 pixels. Next luminance is extracted from each RGB image to create gray-scale images. We sample N points, here $N = 77$, at $(20j, 20k)$ for $(j = 1 \ldots 11, k = 1 \ldots 7)$ on an image. Let (x_i, y_i) denote each sample point on the previous frame image for $i = 1, \ldots N$ and (x'_i, y'_i) denote corresponding point on the current frame image. Optical flow vector (u_i, v_i) of each sample point i is given

by

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} x'_i \\ y'_i \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \end{pmatrix}. \quad (2)$$

The mean (μ_x, μ_y) of N optical flow vectors,

$$\begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix} = \begin{pmatrix} \frac{\sum_{i=0}^{N} u_i}{N} \\ \frac{\sum_{i=0}^{N} v_i}{N} \end{pmatrix}, \quad (3)$$

represents a camera shift, such as pans or tilts. The standard deviation (σ_x, σ_y) for camera shift is calculated as follows:

$$\begin{pmatrix} \sigma_x \\ \sigma_y \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{\sum_{i=0}^{N}(u_i-\mu_x)^2}}{N} \\ \frac{\sqrt{\sum_{i=0}^{N}(v_i-\mu_y)^2}}{N} \end{pmatrix}. \quad (4)$$

In addition, we define zoom ratio z as

$$z = \frac{\sum_{i=1}^{N}(u_i-\mu_x)(x'_i-x_c) + (v_i-\mu_y)(y'_i-y_c)}{N}, \quad (5)$$

where x_c, y_c denote X-Y coordinates at the center of the current image. The five features, $(\mu_x, \mu_y, \sigma_x, \sigma_y, z)$, are used as camera motion features.

4. Language model

We apply a language model that is widely used in the speech recognition field to represent scene contexts for scene recognition. There are two main types of language models, grammar-based language models and statistical language models (SLMs). Grammar-based language models are a formal specification of the permissible structures for the language, whereas SLMs model the probabilistic relationship among the sequence of words. A grammar-based language model is inappropriate for scene recognition of sports broadcasts because unexpected scenes occur frequently, and skipped scenes also happen sometimes. Therefore, we apply *n*-gram models, one kind of SLMs.

SLM calculates the appearance probability $P(s_1 \dots s_N)$ from a given scene sequence $s_1 \dots s_N$. Let s_1^N denote the scene sequence $s_1 \dots s_N$. The appearance probability $P(s_1 \dots s_N)$ is calculated as follows

$$P(s_1^N) = \prod_{i=1}^{N} P(s_i \mid s_1^{i-1}). \quad (6)$$

A large mount of training data is required for estimating the appearance probability of a long scene sequence. To avoid this problem, n-gram models are used. In n-gram models, it is assumed that the probability of a current scene depends on only $n-1$ previous scenes:

$$P(s_1^N) = \prod_{i=1}^{N} P(s_i \mid s_{i-n+1}^{i-1}). \quad (7)$$

The models for $n = 1, 2, 3$ are called unigram, bigram, and trigram models, respectively. The conditional probability in Eq. (7) is calculated by maximum likelihood estimation (MLE) using training data. For example, the trigram probability is calculated as follows.

$$P(s_i \mid s_{i-2}^{i-1}) = \frac{C(s_{i-2}^{i})}{C(s_{i-2}^{i-1})}, \quad (8)$$

where $C(s_{i-2}^{i})$ denotes the number of appearances of scene sequence s_{i-2}^{i}. In Eq. (8), unobserved sequences of scenes in the training data are given the probability zero. In addition, the probability for those n-grams that occur only a few times in the training data might not be estimated correctly. To solve these data sparseness problems, several smoothing techniques are used in which the probabilities of unobserved n-grams are approximated by using those of lower order n-grams. Here, we employ Witten-Bell smoothing, which smoothes based on the number of types of scenes that occur after a certain scene sequence [18, 19]. Let $R(s_{i-2}^{i-1})$ denotes the number of scene types that occur after s_{i-2}, s_{i-1}. We calculate the trigram probability as follows:

$$P(s_i \mid s_{i-2}^{i-1}) = \begin{cases} \frac{C(s_{i-2}^{i})}{C(s_{i-2}^{i-1})+R(s_{i-2}^{i-1})} & \text{if } C(s_{i-2}^{i}) > K, \\ \frac{R(s_{i-2}^{i-1})}{C(s_{i-2}^{i-1})+R(s_{i-2}^{i-1})} \alpha P(s_i \mid s_{i-1}) & \text{else if } C(s_{i-2}^{i-1}) > 0, \\ P(s_i \mid s_{i-1}) & \text{otherwise}, \end{cases} \quad (9)$$

where α denotes a normalization coefficient to make the total probability become one, and K denotes a cut-off threshold. A cut-off is a procedure that removes n-grams whose number of appearances is less than a certain threshold, in order to reduce the number of n-grams.

Table 1: Scene type, label and number of appearances in 5 groups, in which "pickoff (po)" includes pickoff throw, "walk (wk)" includes being hit by a pitch, "steal (st)" includes being caught stealing, "out of play (op)" refers to game scenes with no play action, such as between innings, and "CG effect (ef)" refers to scenes with CG effect, such as batting average of a baseball player.

Scene type	Label	Group ID 1	2	3	4	5	Total
ball	b	271	371	353	308	398	1701
replay	rp	235	351	270	356	366	1578
strike	s	192	192	221	194	263	1062
out of play	op	185	197	181	199	175	937
foul	f	160	166	199	200	170	895
ground out	go	78	67	73	86	76	380
fly out	fo	70	81	66	60	75	352
CG effect	ef	58	72	52	52	38	272
strike out	so	41	36	48	49	49	223
base hit	bh	38	49	38	41	37	203
pickoff	po	24	19	25	35	36	139
walk	wk	25	38	23	23	27	136
clutch hit	ch	6	17	10	9	17	59
extra-base hit	ebh	10	9	12	11	8	50
home run	hr	9	9	4	10	6	38
steal	st	4	4	6	9	1	24

5. Experiments

5.1. Experimental conditions

We used 25 games (75 hours) of baseball broadcast video provided by NHK (Japan Broadcasting Corporation) Science & Technical

Research Laboratories. We applied 16 scene labels (Table 1). We divided the 25 games into five groups. Recognition experiments were carried out by cross-validation in which the video data of one group was used as test data and those of the other four groups were used as training data. The results were averaged over those five groups. Table 1 shows the number of appearances for each scene in these five groups.

We used *precision* P and *recall* R for the evaluation of scene recognition. For each scene label l, precision and recall are calculated as

$$P = \frac{C}{S}, \qquad R = \frac{C}{T}, \tag{10}$$

where C is the number of frames that were correctly recognized as label l, S is the number of frames that were recognized as label l (including incorrect recognition results) , and T is the number of frames that represents label l. In some applications, precision is more important than recall, but in others, recall is more important. However, for most applications, it is desirable that both precision and recall are high. From this viewpoint, we also used *F-measure*, a harmonic average of precision and recall:

$$F = \frac{2PR}{P+R}. \tag{11}$$

In the training phase, a scene HMM was prepared for each scene label (Table 1) by using the Hidden Markov Model Toolkit (HTK) [20]. All the scene HMMs had the same number of states, 30, which was optimized in our preliminary experiment. The features used in this experiment were the three features explained in Section 3.4: low frequency features (LFs), dynamic low frequency features (DLFs), and camera motion features (CFs). Training of bigram and trigram models was carried out by using CMU-SLM-Toolkit [21], where the cut-off threshold was zero and Witten-Bell smoothing was used. Recognition experiments were carried out for the test data using the prepared video model and a language model. Julius, an open source, real-time large vocabulary speech recognition engine, was used as a decoder [22].

5.2. Experimental result

Table 2 shows the result of the baseline, bigram, and trigram models. The baseline was a simple grammar used in our previous work and explained in Section 3.1. Table 2 indicates that using scene contexts was effective for scene recognition. In particular, our proposed method was effective for recognition of replay scenes. Replay scenes often appear after highlight scenes such as home runs. Therefore, some recognition errors of replay scenes that occurred after highlight scenes were reduced by using n-gram models. However, the results of recognition performance of some scenes, such as CG effect, strike, pickoff, and ground out, deteriorated slightly. This might be because scene sequences that included these scenes in test data hardly ever occurred in training data. This problem can most likely be solved by increasing the training data. The results of recognition performance using the trigram model had slightly better performance than that of the bigram model. This might be because the trigram model represented longer scene contexts. The recognition performance of steal, base hit, extra-base hit, and clutch hit were still low. The insufficient result of steal was caused by a limited number of samples, as shown in Table 1. The insufficient result of base hit, extra-base hit, and clutch hit scenes was caused by confusions among each other, ground out, and fly out scenes. This might be due to the fact that our proposed features do not represent differences among these scenes well enough. There was much confusion among strike, ball, and foul scenes as well. Users' demands for extraction of these scenes are, however, relatively low. When we ignore the confusion among those three scenes, the F-measure average over all scenes using the baseline, bigram, and trigram models were 45.0%, 48.1%, and 48.6%, respectively.

6. Conclusions

This paper has proposed a new scene recognition framework using scene context information, and applied the proposed method to baseball broadcast video.

In this framework, HMMs and n-gram models are used to model each scene and scene context, respectively. Three features, that is, low frequency features (LFs), dynamic low frequency features (DLFs), and camera motion features (CFs), are used as input features. In our evaluation using 25 games, the F-measure was improved by 3.3% when the trigram model was used.

The proposed method is also expected to be applicable to other video sports contents, such as basket ball and American football, and to broadcast news, where it is relatively easy to define scenes and model scene contexts.

This study should be regarded as the starting point of scene recognition using a statistical language model to represent scene contexts. Many problems still remain to be solved. First, we should compare the proposed method with conventional rule-based methods to prove the effectiveness of our method. Second, we should improve the language model. A combination of the statistical language models and the rule-based grammars is promising, and we also need more training data to improve the language models. Third, we should explore new video models and features to improve performance. Finally, we plan to extend our framework to a *multi-modal* recognition framework that deals not only with a video mode but also with other modes, such as speech and text.

7. References

[1] R. Brunelli, O. Mich, and C. M. Modena, "A survey on the automatic indexing of video data," *Journal of Visual Communication and Image Representation*, vol. 10, no. 2, pp. 78–112, 1999.

[2] S. Kumar and M. Hebert, "A hierarchical field framework for unified context-based classification," *Proc. IEEE International Conference on Computer Vision*, vol. 3, pp. 1284–1291, 2005.

[3] H. B. Nguyen, K. Shinoda, and S. Furui, "Robust highlight extraction using multi-stream hidden Markov models for baseball video," *Proc. IEEE International Conference on Image Processing*, vol. 3, pp. 173–176, 2005.

[4] T. Mochizuki, M. Tadenuma, and N. Yagi, "Baseball video indexing using patternization of scenes and hidden Markov model," *Proc. IEEE International Conference on Image Processing*, vol. 3, pp. 1212–1215, 2005.

[5] P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden Markov models," *Proc. IEEE International Conference on Image Processing*, vol. 1, pp. I–609–612, 2002.

[6] Y. Gong, M. Han, W. Hua, and W. Xu, "Maximum entropy model-based baseball highlight detection and classification,"

Table 2: Results of precision (P(%)), recall (R(%)), and F-measure (F(%)) using baseline, bigram, and trigram models, in which baseline is a simple grammar, and bigram and trigram models are the proposed method. Bigram and trigram models were constructed where cut-off threshold was zero, and Witten-Bell smoothing was used.

	Baseline			**Bigram**			**Trigram**		
Scene	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
hr	52.3	41.3	46.1	68.0	38.0	48.8	77.0	42.0	54.4
ch	25.1	22.5	24.0	38.9	22.9	28.8	38.5	21.4	27.5
ebh	12.4	20.4	15.4	23.0	16.7	19.4	28.3	18.8	22.6
bh	28.0	35.8	31.4	28.1	36.6	31.8	33.5	37.2	35.3
wk	42.2	32.5	36.7	48.4	34.2	40.1	45.1	35.6	39.8
st	4.7	2.9	3.5	11.6	2.9	4.6	14.8	2.9	4.8
po	40.0	54.4	46.1	42.2	51.7	46.5	40.9	49.2	44.7
s	31.3	49.6	38.4	32.2	45.3	37.6	31.5	39.2	34.9
b	49.7	34.0	40.4	48.3	43.1	45.6	48.5	47.2	47.8
f	46.6	37.9	41.8	44.1	42.8	43.4	44.0	43.2	43.6
so	48.6	63.2	54.9	60.4	45.5	51.9	61.0	52.3	56.3
fo	44.8	47.8	46.3	44.1	50.9	47.3	47.8	50.3	49.0
go	53.2	60.6	56.7	50.7	56.8	53.6	51.8	59.8	55.5
ef	95.0	86.9	90.8	96.0	85.7	90.6	96.1	81.8	88.4
rp	69.8	38.1	49.3	68.3	49.2	57.2	64.2	50.6	56.6
op	32.6	59.8	42.2	45.9	57.1	50.9	50.3	57.2	53.5
average	42.3	43.0	41.5	46.9	42.5	43.6	48.3	43.0	44.7

International Journal of Computer Vision and Image Understanding, vol. 96, pp. 181–199, 2004.

[7] C.-H. Liang, W.-T. Chu, J.-H. Kuo, J.-L. Wu, and W.-H. Cheng, "Baseball event detection using game-specific feature sets and rules," *Proc. IEEE International Symposium on Circuits and Systems*, pp. 3829–3832, 2005.

[8] P. Xu, L. Xie, S. F. Chang, A. Divakaran, A. Vetro, and H. Sun, "Algorithms and system for segmentation and structure analysis in soccer video," *Proc. IEEE International Conference on Multimedia and Expo*, pp. 928–931, 2001.

[9] Y. Gong, L.-T. Sin, C.-H. Chuan, H.-J. Zhang, and M. Sakauchi, "Automatic parsing of TV soccer programs," *Proc. IEEE International Conference on Multimedia Computing and Systems*, pp. 167–174, 1995.

[10] E. Kijak, L. Oisel, and P. Gros, "Hierarchical structure analysis of sport videos using HMMs," *Proc. IEEE International Conference on Image Processing*, vol. 3, pp. 1025–1028, 2003.

[11] G. Xu, Y.-F. Ma, H.-J. Zhang, and S.-Q. Yang, "Motion based event recognition using HMM," *IEEE Trans. Circuits and Systems*, vol. 15, pp. 1422–1433, 2005.

[12] N. Babaguchi, Y. Kwai, and T. Kitahashi, "Event based indexing of broadcasted sports video by intermodal collaboration," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 68–75, 2002.

[13] L. Rabiner and B.-H. Juang, "Fundamentals of speech recognition," *Prentice Hall*, 1993.

[14] G. Xu, Y.-F. Ma, H.-J. Zhang, and S. Yang, "Motion based event recognition using HMM," *Proc. IEEE International Conference on Pattern Recognition*, vol. 2, pp. 831–834, 2002.

[15] D. Zhong and S. F. Chang, "Structure analysis of sports video using demain models," *Proc. IEEE International Conference on Multimedia and Expo*, pp. 920–923, 2001.

[16] S. Takagi, S. Hattori, K. Yokoyama, A. Kodate, and H. Tominaga, "Sports video categorizing method using camera motion parameters," *Proc. IEEE International Conference on Multimedia and Expo*, pp. 461–464, 2003.

[17] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. 7th International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

[18] P. Placeway, R. Schwartz, P. Fung, and L. Nguyen, "The estimation of powerful language models from small and large corpora," *Proc. IEEE Acoustics, Speech and Signal Processing*, vol. II, pp. 33–36, 1993.

[19] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Trans. Information Theory*, vol. 37, no. 4, pp. 1085–1094, 1991.

[20] "HTK," `http://htk.eng.cam.ac.uk`.

[21] "CMU SLM Toolkit," `http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html`.

[22] "Julius," `http://julius.sourceforge.jp`.