DETECTING HIGHLIGHTS IN SPORTS VIDEOS: CRICKET AS A TEST CASE

Hao Tang[1,2]*, Vivek Kwatra*[2]*, Mehmet Emre Sargin*[2]*, Ullas Gargi*[2]

[1]HP Labs, Palo Alto, CA USA
[2]Google Inc., Mountain View, CA USA

ABSTRACT

In this paper, we propose a novel approach for detecting highlights in sports videos. The videos are temporally decomposed into a series of events based on an unsupervised event discovery and detection framework. The framework solely depends on easy-to-extract low-level visual features such as color histogram (CH) or histogram of oriented gradients (HOG), which can potentially be generalized to different sports. The unigram and bigram statistics of the detected events are then used to provide a compact representation of the video. The effectiveness of the proposed representation is demonstrated on cricket video classification: Highlight *vs.* Non-Highlight for individual video clips (7000 training and 7000 test instances). We achieve a low equal error rate of 12.1% using event statistics based on CH and HOG features.

***Index Terms*—** highlight detection, sports video, event discovery, event detection, video clip representation

1. INTRODUCTION AND RELATED WORK

Sports video analysis has been an attractive research area in the multimedia community [1]. Sports videos appeal to a large population of people all around the world, and have become an important form of multimedia content that is streamed over the Internet and television networks. Every single day, tens of millions of people watch sports videos of various kinds, including soccer, cricket, tennis, baseball, basketball, etc., just to name a few. Normally, sports videos are rather long, consisting of portions which are interesting or exciting and portions which are boring, bland, and likely "a waste of the viewer's valuable time." If possible, most viewers prefer to watch only the interesting or exciting portions of the videos, and would rather skip the boring parts. Therefore, automatic detection of these *highlights* in sports videos has become a fundamental problem of sports video analysis, and is receiving increased attention from researchers in the field [2, 3, 4, 5, 6, 7, 8].

The task of highlight detection in sports videos is aimed at automatically assigning a label to a particular segment or clip of a sports video that indicates whether the video segment or clip is interesting (*i.e.* a highlight clip) or not (*i.e.* a non-highlight clip). Most previous methods for highlight detection in particular, and for sports video analysis in general, are based on mid-level and high-level audio-visual features, such as player trajectories, crowds, audience cheering, goal or score events, etc., and primarily focus on certain kinds of sports videos such soccer, baseball or basketball videos [9, 10]. For example, representative works on highlight detection extract high-level events such as the score event in soccer games [2], the hit event in baseball games [3], and the goal event in basketball games [4]. This is a challenging task, and many methods rely on detecting audio events in the video where audience cheering provides the most convincing cue for highlight detection. The mid-level and high-level audio-visual features are usually difficult to extract from the videos robustly, and are likely to be specifically designed to cope with videos of a particular sport. Once these features are extracted from the videos, certain heuristic rules or statistical models are developed for the particular sport based on its inherent structural constraints, as defined by the rules of the game and field production. Such methods may work well for the sport (or the specific high-level events) for which they are designed, but can be difficult to generalize. Furthermore, mid-level and high-level features carry certain semantic meanings in the context of the sport, the extraction of which is still an open research issue.

In this paper, we propose a novel approach for detecting highlights using easy-to-extract low-level visual features such as the color histogram (CH) [11] or histogram of oriented gradients (HOG) [12]. In particular, we focus on cricket, which is an outdoor bat-and-ball team sport similar to baseball. It is played professionally in many countries, and has become the world's second most popular sport after soccer. Even though our methodology does not use sport-specific features, we have chosen cricket as a test case in part due to the availability of a large labeled dataset of video clips from a cricket tournament. There has been prior work on cricket highlights generation [8] based on domain-specific modeling of semantic concepts and events, but it is highly customized for cricket.

Our approach is based on an unsupervised event discovery and detection framework, which leads to an effective video clip representation composed of the unigram and bi-

THIS WORK WAS DONE WHEN THE FIRST AUTHOR WAS A SUMMER INTERN AT GOOGLE INC., MOUNTAIN VIEW, CA.

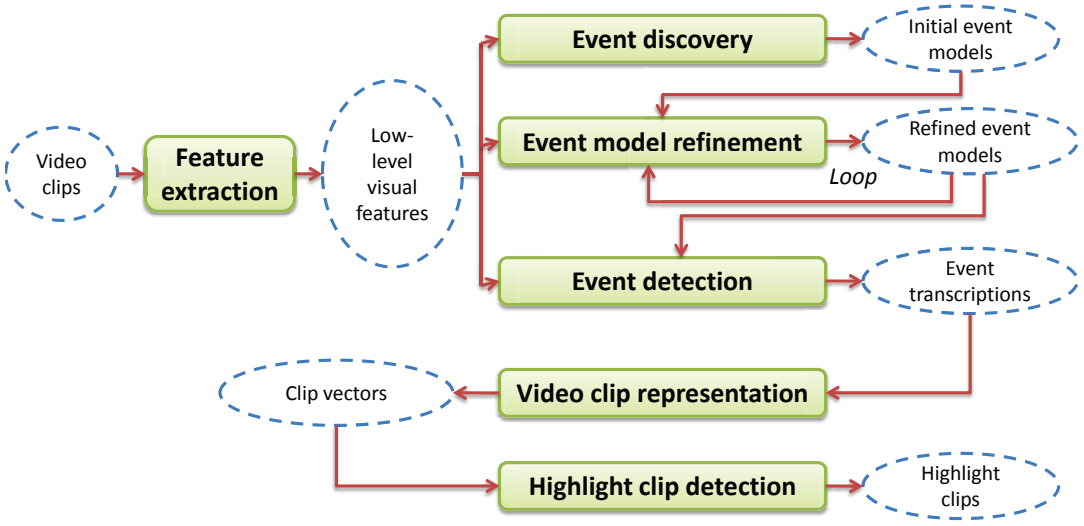

Fig. 1: The overview diagram of our proposed approach.

gram statistics of the detected events. Once the video clips have been transcribed using the discovered events, the supervised phase trains a linear support vector machine (LSVM) classifier [13] from clips labeled as highlight or non-highlight. We achieve a low equal error rate of 15.7% using CH features, 12.6% using HOG features, and 12.1% by combining CH and HOG features, on a cricket video data set consisting of 7000 training clips and 7000 test clips.

2. OVERVIEW

Our proposed approach is schematically illustrated in Figure 1. Before we proceed to the details of the approach, let us introduce the concept of an event which is used throughout the paper. An event in the scope of this paper is defined as a short coherent temporal pattern in a video clip, which does not necessarily have anything to do with high-level semantics and which may be identified solely based on low-level visual features. As shown in the overview diagram in Figure 1, our approach consists of six stages: feature extraction, event discovery, event model refinement, event detection, video clip representation, and highlight clip detection. Feature extraction is used to pre-process all videos; event discovery and model refinement stages construct the event vocabulary; event detection and video clip representation transcribe the video; and finally feed into the training of the highlights detection classifier.

At the feature extraction stage, frame-based low-level visual features such as the color histogram (CH) or histogram of oriented gradients (HOG) are extracted from the video clips using relevant image processing and computer vision algorithms. At the event discovery stage, we determine a set of events out of many video clips based on the extracted low-level visual features through a process known as diarization (a.k.a. segmentation and clustering) [14], and learn a corresponding set of initial event models using hidden Markov models (HMMs) [15] and the embedded training technique [16]. At the event model refinement stage, all video clips are transcribed into event sequences based on the set of initial event models via Viterbi decoding [17], and the generated event transcriptions are in turn used to learn a new set of event models. In general, the likelihood of event models increases after re-training. Therefore, the new set of event models is said to be a set of refined models. This model refinement process can be performed for several iterations until the likelihood of the event models no longer increases, leading to a set of final event models. At the event detection stage, the set of final event models are used to decode an arbitrary test video clip into a sequence of events; at which point, the video clip is represented as a sequence of events from the discovered models. At the video clip representation stage, we form a clip vector for each input video clip, which is composed of the unigram and bigram statistics of the events in the event sequence detected from the video clip. Finally, at the highlight clip detection stage, we use a linear support vector machine (LSVM) classifier to classify a given video clip into one of the two categories, highlight or non-highlight, in the clip vector representational space.

We would like to emphasize that all the stages prior to highlight clip detection (*i.e.* up to the construction of a clip vector for a test video clip) form an unsupervised event discovery and detection framework, which leverages low-level visual features to transcribe a video clip into an event sequence, and further offers an effective video clip representa-

tion based on the unigram and bigram statistics of the detected events. Even though we limit our experiments to cricket, this framework is generic in nature and depends primarily on low-level visual features; therefore in principle it can be applied to videos of other sports. In the following, we describe the individual elements of our approach in detail.

3. LEARNING AND DETECTION PIPELINE

3.1. Feature extraction

Frame-based low-level visual features may be extracted from a video clip with ease using mature image processing and computer vision techniques. Some commonly seen examples of low-level visual features are color histogram (CH), histogram of oriented gradients (HOG), histogram of oriented optical flow (HOOF) [18], and so forth. In this work, we simply adopt CH features and HOG features, as they can be computed from a video clip very quickly. Although more sophisticated features such as HOOF features do contain motion information, the incorporation of which may further lead to increased performance, in our experiments, both CH and HOG features have yielded surprisingly good highlights detection rates as a result of using our proposed approach.

3.2. Event discovery

As mentioned earlier, within the scope of this paper, an event is defined as a short coherent temporal pattern in a video clip which can be identified from low-level visual features alone. There is a clear motivation for this concept. An event, by definition, represents a short sequence of consecutive video frames with coherent and compact support in the space of low-level visual features, but does not necessarily carry any semantic meaning regarding the video content. However, the high-order co-occurrence relationships between these events may provide useful cues for discriminating between highlights and non-highlights in a video.

Given many video clips, we discover a set of events through a process known as diarization (a.k.a. segmentation and clustering, for the reason that follows immediately). The diarization process is essentially a two-step procedure: first segmentation and then clustering. It is preferred that diarization is performed shot-wise. That is, for any video clip, we first preprocess it into individual shots using a color histogram based shot boundary detection algorithm. Within each shot, we segment the video frames into small chunks that last 500 milliseconds. Note that we purposely over-segment the video frames at this point so we can assume that a single chunk corresponds to only a single event (namely there is no event change point within the chunk). We then cluster these small atomic chunks using a bottom-up hierarchical agglomerative clustering algorithm with the Ward's linkage function [19], which specifies the distance between two clusters and which is computed as the increase in the "error sum of squares" (ESS) after fusing two clusters into a single cluster. Note that at this stage one can alternatively use the K-means clustering algorithm [20], which can produce similar clustering outcomes, but at a much slower speed than the hierarchical clustering algorithm adopted in this work. The result of clustering is a set of discovered events, or event vocabulary, $\mathcal{E} = \{e_1, e_2, \cdots, e_K\}$, and as a by-product, the transcriptions of the video clips in terms of the discovered events, $\mathcal{T}_i = e_{i1}, e_{i2}, \cdots, e_{iL_i}$, where $e_{il} \in \mathcal{E}$.

Since an event is a dynamically evolving pattern over a short time span, it is ideally suited to be described using a Hidden Markov Model (HMM) to capture the temporal dynamics. Thus, we learn a set of K HMMs, each for one of the K events discovered, from the generated event transcriptions of the video clips using the embedded HMM training technique. Contrary to the conventional training technique that trains the event HMMs independently, the embedded training technique first concatenates the event HMMs to form a clip HMM according to the event transcription, and then trains the inter-connected event HMMs simultaneously. A great advantage of the embedded training technique over the conventional training technique is that we only need event labels in the event transcription. There is no time boundary information of the individual events required. The embedded training eventually leads to a set of initial event models, $\mathcal{M}^I = \{m_1^I, m_2^I, \cdots, m_K^I\}$.

3.3. Event model refinement

The set of initial event models $\mathcal{M}^I$ learned at the event discovery stage is coarse, because the generated event transcriptions of the video clips as a result of the diarization process is rather noisy. This is primarily due to the lack of the consideration of inter-frame statistical dependence. At this stage, we introduce an event model refinement method, which can improve the quality of the event models in an iterative manner. The model refinement method proceeds as follows: the initial event models are used to decode the video clips into event transcriptions, which are in turn used to re-train a set of new event models using the HMM embedded training technique. This procedure generally yields an increase in the likelihood of the event models upon convergence of the training algorithm. In this sense, we say that the set of new event models has been refined from the set of initial event models. It is advisable that we repeat this procedure for multiple iterations until the likelihood of the event models no longer increases. At this point, we have learned a set of final event models, $\mathcal{M}^F = \{m_1^F, m_2^F, \cdots, m_K^F\}$, ready to be used to perform event detection from novel video clips.

3.4. Event detection

In our approach, event detection is nothing special but transcribing an input video clip into a sequence of events based on the set of final event models $\mathcal{M}^F$ obtained at the previous

Fig. 2: Three examples of the discovered events. Each row corresponds to four key frames of the video segment of an event.

stage. This is achieved by the Viterbi decoding algorithm for HMMs, which uses the dynamic programming technique to find the maximum likelihood path in a time-event lattice that represents the most likely sequence of events, given the event models and a video clip. Note that during the event discovery stage and event model refinement stage, we only use a limited number of video clips, which we refer to as the training set (but remember that both stages are completely unsupervised), to discover the events and to learn the event models. Once the set of final event models are obtained, we may use it to decode an arbitrary input video clip into a sequence of events, such as one in a test set.

3.5. Video clip representation

The above stages generate a sequence of events for any input video clip. At this stage, we further form a single-vector representation of a video clip which is composed of the unigram and bigram statistics of the detected events. Specifically, the unigram statistics of the detected events is given by the histogram of the individual discovered events (*i.e.* a K-vector), which characterizes how likely the events happen individually in the video sequence. The bigram statistics of the detected events is given by the histogram of event pairs (*i.e.* a K^2-vector), which characterizes how likely the pairs of events co-occur (one after another) in the video sequence. Both histograms are normalized and concatenated to form a clip vector of dimension $K + K^2$, which turns out to be an effective representation for the purpose of highlight detection.

3.6. Highlight clip detection

The problem of detecting highlight video clips is formulated as a binary classification problem in the clip vector representational space. Based on a training set with numerous positive examples (highlight clips) and negative examples (non-highlight clips), each represented by a clip vector, a linear support vector machine (LSVM) classifier is trained to classify input video clips (in the form of a clip vector) into one of two categories: highlight clips or non-highlight clips.

4. EXPERIMENTS

Our experiments are carried out on a large dataset of cricket videos, consisting of 7000 training clips and 7000 test clips. Altogether, the 14000 clips correspond to 60 cricket matches (games) from a 20 *overs* a side tournament (1 *over* $=$ 6 *balls*, averaging $\approx$ 240 balls per match). Each video clip corresponds to a single ball in a cricket match and typically lasts for 10 to 20 seconds. It is manually labeled by a human as an *N-run*, where $N \in \{1, 2, 3, 4, 5, 6\}$, or a *wicket*. By rule, those video clips corresponding to either a *4-run*, a *6-run*, or a *wicket* are considered as highlight clips and all the other video clips are considered as non-highlight clips. The dataset is split into training and test by using 30 complete matches for training clips and the remaining 30 for test clips.

Following the pipeline in Figure 1, we extract both CH features and HOG features from every frame of all the 14000 video clips. We notice that the dimension of the raw CH or HOG feature vectors are very high (of the order of thousands). In order to alleviate the "curse of dimensionality" and sup-

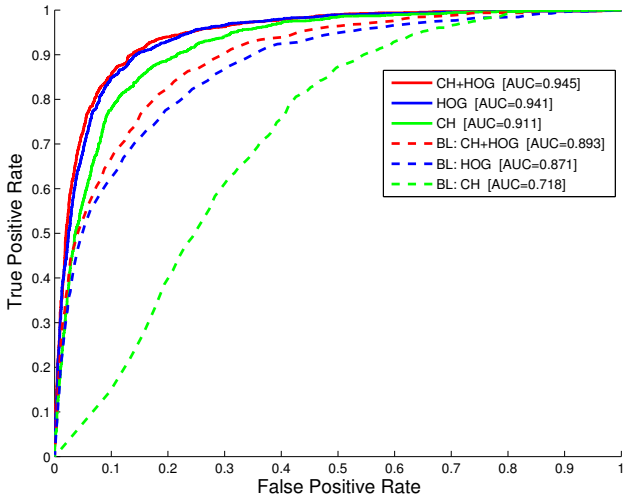

Fig. 3: ROC curve for highlights detection using CH, HOG, and CH+HOG features. Dashed curves (marked BL) correspond to baseline results.

press the noise in the features, we learn a linear projection matrix via principal component analysis (PCA) based on the 7000 video clips in the training set. The learned linear projection is used to reduce the dimension of CH or HOG feature vectors to 50 while preserving a sufficiently large percentage of the energy of the original feature vectors. In the projected feature vector space, for each of the two cases (CH and HOG), the 7000 video clips in the training set are first used to discover a set of K events and to learn a corresponding set of K final event models, as described in Section 3. Note that in our experiments, the number of events K is empirically chosen to be 30. Larger values of K do not significantly increase the detection performance. Subsequently, all 14000 video clips in both the training and test sets are decoded into 14000 event transcriptions by the Viterbi algorithm based on the set of 30 final event models. Using these 14000 event transcriptions, we finally construct the corresponding 14000 clip vectors, each of which has a dimension of $30 + 30^2 = 930$.

The 7000 clip vectors derived from the training set are used to learn an LSVM classifier, which is then used to perform binary classification of highlight and non-highlight video clips on the 7000 clip vectors derived from the test set. Since in both the training and test sets, the number of highlight clips *vs.* non-highlight clips are highly unbalanced (roughly 1:4), we evaluate the detection performance by continuously modulating the decision threshold of the LSVM classifier, thereby obtaining the receiver operating characteristics (ROC) curve, and the area under this curve (AUC) and its equal error rate (EER) as performance measures.

Figure 2 shows a few examples of the discovered events. For each event (a row in the figure), we only display four key frames of the video segment, as there is no easy way to visualize the entire video segment. By viewing the video segments of all the discovered events, we observe that the video frames within an event do exhibit consistent visual patterns while the video frames across different events show largely varied visual appearances.

Our first experiment is based on CH features, which can be easily and quickly extracted from the video clips. The color histogram is a representation of the distribution of colors in an image, which can in principle be built for any kind of color space. In this work, we adopt the commonly used three-dimensional RGB color space to form 3D color histograms. One well-known drawback of CH features is that being low-level descriptors of images, CH features are solely dependent of the global color property of the video, without taking into consideration any shape and texture properties of objects in the video. Nonetheless, the highlight clip detection performance of our approach based on CH features is reasonable. Solid green curve in Figure 3 shows the ROC for this experiment. The AUC is computed as 0.911 (AUC=1.0 being perfect detection) and EER is 15.7% (see Table 1). The good performance, despite the fact that very little information regarding the structure of the video content is carried by the low-level CH features, is attributable to our robust unsupervised event discovery and detection framework, which leads to an effective video clip representation.

Table 1: Equal error rates (EERs) for the two experiments.

Type of features	Equal Error Rate
CH	15.7%
HOG	12.6%
CH+HOG	12.1%

Our second experiment is based on HOG features. Unlike CH features, which extract color information, HOG features extract shape and texture information. However, HOG features are also low-level descriptors of images and do not carry much information regarding the temporal structure of the video content. The highlight detection result based on HOG features outperforms the result based on CH features. Solid blue curve in Figure 3 shows the ROC for this experiment. The AUC is computed as 0.941, which is a 3% absolute increase compared to CH features. Similarly EER reduces to 12.6%, a 3.1% improvement over CH.

We also conducted a third experiment by combining the CH and HOG based features. We only combine them in the last stages of our algorithm by concatenating the video clip representations (unigram and bigram histograms) corresponding to each feature type. We then use this joint representation to train the LSVM classifier. The resulting ROC curve is plotted as the solid red curve in Figure 3 with AUC=0.945 and EER=12.1%, which is a marginal improvement over using HOG alone. Altogether, these results indicate that HOG features result in more discriminative clip representations than CH, which is to be expected: HOG features are obtained

by evaluating normalized local histograms on a dense grid, and are therefore less sensitive to illumination and geometric transformations, *e.g.* HOG features can discriminate objects with different shape and texture in a video, in the same way they are used for human and object detection tasks in images.

Finally, we compare our technique with a baseline approach based on standard HMMs: Features extracted from highlight and non-highlight videos in the training set are used to train two HMMs with 30 states, where the transition matrix is assumed to be fully connected. The observation distributions are modeled with a single Gaussian where the covariance matrix is assumed to be diagonal. Given two HMMs for highlight and non-highlight, the posterior probability of a test clip being highlight is obtained using the Bayes rule assuming uniform prior. CH and HOG features are used to obtain two posterior probabilities which are combined using the product rule with exponential weights. The optimum weight combination is found by maximizing the AUC measure on the training set. The dashed curves in Figure 3 show ROC plots for baseline results (colors match with the solid curves for each feature type). In each case, our event discovery based technique outperforms the HMM-only baseline approach.

5. CONCLUSION

In this paper, we propose a novel approach for detecting highlights in sports videos using easy-to-extract low-level visual features such as the color histogram (CH) or histogram of oriented gradients (HOG). Our approach is based on a robust unsupervised event discovery and detection framework which immediately leads to an effective sports video clip representation composed of the unigram and bigram statistics of the detected events. Using a linear support vector machine classifier, we achieve a low equal error rate of 12.1% using CH and HOG features on a cricket video data set consisting of 7000 training clips and 7000 test clips. The attractiveness of our approach lies in the fact that it is based on easy-to-extract, domain-independent low-level visual features, and therefore can potentially be generalized to deal with videos of other sports, something we wish to explore as part of future work. Another promising future work direction is to improve detection performance by incorporating low-level visual features that characterize motion information in the video, such as the histogram of oriented optical flow (HOOF).

6. REFERENCES

[1] J.R. Wang and N. Parameswaran, "Survey of sports video analysis: research issues and applications," in *Proc. Pan-Sydney area workshop on visual information processing*, Darlinghurst, Australia, June 2004, p. 8790.

[2] D. Yow, B.L. Yeo, M. Yeung, and B. Liu, "Analysis and presentation of soccer highlights from digital video," in *ACCV*, Singapore, December 1995, pp. 1–6.

[3] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for tv baseball programs," in *ACM Multimedia*, Los Angeles, CA, October 2000, pp. 105–115.

[4] S. Nepal, U. Srinivasan, and G. Reynolds, "Automatic detection of 'goal' segments in basketball videos," in *ACM Multimedia*, Ottawa, Canada, October 2001, pp. 261–269.

[5] A. Hanjalic, "Adaptive extraction of highlights from a sport video based on excitement modeling," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1114–1122, 2005.

[6] J. Wang, C. Xu, E. Chng, and Q. Tian, "Sports highlight detection from keyword sequences using hmm," in *IEEE ICME*, Taipei, China, June 2004, pp. 599–601.

[7] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T.S. Huang, "Highlights extraction from sports video based on an audio-visual marker detection framework," in *IEEE ICME*, Amsterdam, The Netherlands, June 2005, pp. 4–7.

[8] Maheshkumar Kolekar and Somnath Sengupta, "Event-importance based customized and automatic cricket highlight generation," *Multimedia and Expo, IEEE International Conference on*, vol. 0, pp. 1617–1620, 2006.

[9] C. Xu, J. Cheng, Y. Zhang, Y. Zhang, and H. Lu, "Sports video analysis: semantics extraction, editorial content creation and adaptation," *Journal of Multimedia*.

[10] T. D'Orazio and M. Leo, "A review of vision-based systems for soccer video analysis," *Pattern Recognition*, vol. 43, no. 8, pp. 2911–2926, 2010.

[11] L.G. Shapiro and G.C. Stockman, *Computer Vision*, Prentice Hall, Englewood Cliffs, NJ, 2003.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, San Diego, CA, June 2005, pp. 886–893.

[13] C.C. Chang and C.J. Lin, "LIBSVM: A library for support vector machines," 2001.

[14] S.E. Tranter and D.A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[15] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[16] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P.C. Woodland, *The HTK Book*, Cambridge University Engineering Department, Cambridge, UK, 2006.

[17] G.D. Forney Jr., "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[18] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *IEEE CVPR*, Miami, FL, June 2009, pp. 1932–1939.

[19] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[20] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, January 1967, pp. 281–297.