Segmentation of Intentional Human Gestures for Sports Video Annotation

Graeme S. Chambers, Svetha Venkatesh, Geoff A.W. West, and Hung H. Bui
School of Computing, Curtin University of Technology, Perth, Western Australia
E-mail: {chambegs,svetha,geoff,buihh}@cs.curtin.edu.au

Abstract

We present results on the recognition of intentional human gestures for video annotation and retrieval. We define a gesture as a particular, repeatable, human movement having a predefined meaning. An obvious application of the work is in sports video annotation where umpire gestures indicate specific events. Our approach is to augment video with data obtained from accelerometers worn as wrist bands by one or more officials. We present the recognition performance using a Hidden Markov Model approach for gesture modeling with both isolated gestures and gestures segmented from a stream.

1. Introduction

Characterisation of video content has received significant research attention in recent years. Multimedia and video is seemingly available everywhere, on the Internet, even on mobile phones. To attempt at describing any of this video data, there must be knowledge of the domain of video to be processed and some limitations imposed on the types of scenes that can be analysed. It is inconceivable at this point to process unknown video and provide meaningful description. The work here extends on these delimitations and proposes that video be augmented with other sensor data to provide another means of generating descriptions, thus avoiding the problems associated with image segmentation. In particular, we propose that actors indicating specific events wear accelerometers in the form of wrist bands, allowing the recognition of the gestures performed. We refer to a gesture as a specific, intentional action by a human in which part of the body is moved in a predefined way.

There are many types of video where key actors perform intentional gestures to indicate specific events. Consider sports video for example. Umpires in the game perform gestures to indicate something about a team, a player, or the game. Their actions are significant in terms of what is going on in the game and what meaningful information can be derived. We would easily be able to determine at which time a particular team scored points (or goals), derive statistics on the entire game or each team, or even find the "winning goal". Being able to derive such information enables automatic generation of highlights and more importantly, rich, contextual labeling of video. The key novelty of our approach is the augmentation of accelerometer data for gesture recognition with the goal of semantically labeling video.

2. Background

In the area of sports video, several attempts have been made at meaningful labeling, including specific sports [3, 7] and automatic generation of highlights [5]. These however, do not provide suitable reusable frameworks for recognising events in various types of sports. All assume knowledge of the domain and have heuristics for the sport being processed. If such domain knowledge must be known, we propose an alternative: that the video is annotated whilst it is being recorded. If officials in the game are wearing sensors that allow action recognition, comprehensive information can be derived about the game by analyzing the gestures performed by the officials. Attempting to recognise gestures performed by officials in typical sports video places tremendous requirements on the segmentation techniques. Previous work in vision based gesture recognition has concentrated on synthetic environments where hand positions or motion trajectories can be calculated with relative accuracy.A typical sports video scene has possibly several players nearby the officials and very active camera movement. Attempting to isolate just one official is a difficult task, let alone recognising the gestures performed by that official.

Gesture recognition using other sensors such as accelerometers is not reliant on any segmentation techniques as movement information is provided directly by the sensors. The decreasing size of such sensors is enabling them to be placed in existing devices such as PDAs, providing other modalities for user interaction[4]. Others use fairly small sensor devices for recognising simple axis–based gestures for human computer interaction [2], or more complex devices [6, 1] with a focus on complex hand sign language

Proceedings of the 10th International Multimedia Modelling Conference (MMM'04)
0-7695-2084-7/04 $ 20.00 © 2004 IEEE

recognition. For practicality, none of these complex devices could be worn by sports officials as they are too cumbersome. Officials require very lightweight and unobtrusive devices such as wrist bands.

3. Gestures

Human gestures inherently exhibit large amounts of variation. It would be unlikely that a human would be able to repeat, in succession, an intentional gesture in the exact same way each time. Rate of execution and orientation of limbs are just two of the possible differences between examples of the same gesture. The uncertainty and variability of gesture matches the problem of recognition well to the hidden Markov model (HMM).

The HMM is a probabilistic technique that is suited to stochastic signals such as speech and gesture. It can be viewed as a process which moves between different (hidden) states, emitting observation symbols on transition from one state to another. In our case, the observation symbols correspond to the feature vectors calculated from acceleration data. Each gesture has its own HMM, where the one with the highest likelihood is determined as the model which best suits a given observation sequence. A HMM, $\lambda = (A, B, \pi)$, is specified by three sets of parameters: the state transition probability distribution A, the observation symbol probability distribution B, and the initial state probability distribution π. In our case, B is a Gaussian mixture, specifying the probabilities of continuous valued features.

Our accelerometers measure acceleration in two orthogonal directions. We mount two accelerometers orthogonal to each other, thus acceleration is measured in 3–D space. The accelerometers are housed in a small wrist watch sized enclosure worn in the form of a wrist band. Obviously the recognition performance of the system could suffer if the band was worn in grossly different orientations on the wrist, thus we treat the band like a watch, where the face of the enclosure is in a similar direction each time the band is worn. The implementation can measure acceleration of up to $\pm 2g$ with 10 bits precision at 184 samples/second. Currently the accelerometers are attached to a prototype board, interfaced with either a Compaq iPAQ PDA or PC serial port. We can represent the acceleration data as either three separate channels of acceleration or use the spherical coordinate system, where $r = \sqrt{x^2 + y^2 + z^2}$ is the magnitude of the acceleration, $\theta = tan^{-1}\left(\frac{y}{x}\right)$ is the azimuthal angle in the xy plane, and $\phi = cos^{-1}\left(\frac{z}{r}\right)$ is the polar angle from the z axis.

Figure 1 shows the output for each channel of a sensor worn on the right arm of an actor performing an instance of the "leg bye" gesture (from cricket). The actual movement in the gesture is as follows: move the right arm over to the left side of the body while at the same time rotating the torso and lifting the right leg slightly then tap the knee with the right hand three times and finally return the arm to the right side of the body. The three obvious peaks in acceleration correspond to the individual taps of the leg. The variation in the three taps somewhat illustrate the variability of the gesture.

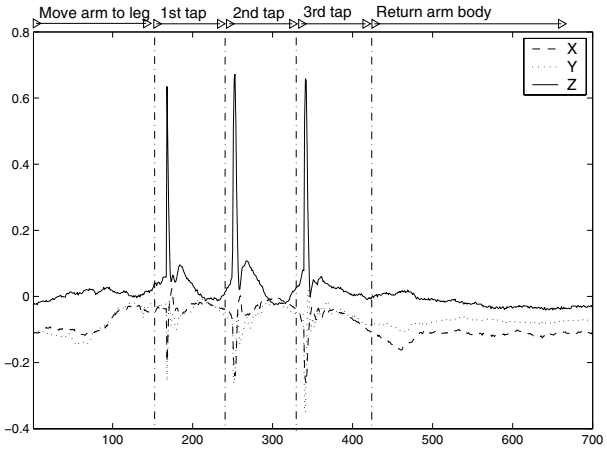

Figure 1. Example "Leg Bye" Gesture

3.1. Segmentation of movement

If gestures are being performed continuously, there needs to be a method for selecting areas of movement to find candidate gestures. Representing the acceleration in spherical coordinate form gives us an advantage for segmenting the acceleration data stream. If the actor is stationary, the acceleration magnitude should be very close to the magnitude of gravity. We can model gravity's magnitude and any very subtle movement using a simple Gaussian distribution. To determine the parameters for the Gaussian, we record several short segments of data where the actor tries not to move. Totaling around 20 seconds, the mean and standard deviation of this training data is calculated and set as the parameters for the Gaussian.

To perform the actual segmentation, a sliding window approach is used for calculating the likelihood of the Gaussian model. For each window of data, the log likelihood of the stationarity Gaussian is calculated. We can use the fact that two adjacent values for log likelihood over a sliding window have some amount of overlap. If we consider the two adjacent log likelihood values and the difference between them, we can say that a sharp change in the log likelihood corresponds to the addition or subtraction of accelerometer data containing movement. Either accelerometer data containing movement is being pushed out of the window to the left or it is being drawn in from the right.

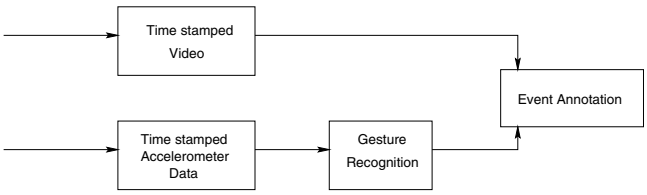

Figure 2. System Overview

3.2. System Overview

Figure 2 shows the architecture of the overall system. As video data is recorded, the movement of key actors wearing the accelerometer wrist bands is also recorded. The accelerometer data is analysed and segmented into candidate gestures then classified accordingly. The video at the time the gesture was performed is then annotated with the event indicated by the gesture. Subsequent analysis of the annotation could be performed enabling richer labeling of the event.

When segmenting candidate gestures from a stream, the window size used is 128 samples with 64 samples overlap. The raw gesture data is first smoothed to filter out any high frequency noise. For both isolated and segmented gestures, features are calculated using a window size of 64 samples with 16 samples overlap. The threshold for movement segmentation by using adjacent log likelihood values is set at 1/128, where the actual log likelihood is normalized by the size of the sliding window.

4. Experiments

To explore the ability of our gesture recognition system, we model a set of umpire gestures from the game of cricket. Umpires in cricket typically stand stationary near the wickets and perform gestures for the audience and other officials to see. We model a set of 10 gestures that are accurately represented by single arm movements, other gestures would require sensors on both arms for recognition.

The cricket umpire gestures we recognise are: *Dead Ball* — sway both arms in front of the body, *Four* — wave the right hand across the body, *Last Hour* — point to the watch on the raised left arm and tap it, *Leg-Bye* — tap the raised right knee, *No Ball* — extend right arm to the side, *One Short* — tap right shoulder with right arm, *Out* — raise arm in front of body with index finger extended, *Penalty Runs* — grasp left shoulder with right hand, *TV Replay* — Outline a rectangle with both hands, and *Wide* — extend both arms out to the sides of the body. Our system is not yet able to handle unknown movements, although this is currently being addressed.

Our dataset consists of a single actor performing all ten gestures on five different days. The dataset has a total of 548 gestures, corresponding to approximately 65 minutes of data. Gestures were captured over different days to introduce variability between the movements. Each day, the sensor was placed on the wrist in approximately the same position and orientation, and the actor performed the gestures in approximately the same manner. The actual gesture movements were mimicked from a real cricket umpire of which we have video footage.

4.1. Isolated Gesture Recognition

For testing the suitability of HMMs in our gesture recognition system, we split the dataset into two and evaluate three different feature sets. Dataset 1 contains all gestures performed on days one through four, and dataset 2 contains all gestures performed on day five. The data is split into two sets to validate the results of the first set with the second. The feature sets are combinations of standard deviation, root mean square, and average vector magnitude, all over the window size for feature calculation. The feature of standard deviation will describe the variation in intensity of acceleration, thus should provide information on regions of sharp or smooth gradual movements. Both average vector magnitude and root mean square will describe the average intensity of acceleration over regions of the gesture. If any of the features are used per acceleration channel, the feature will then be able to indicate the dominant axes of acceleration.

We summarize the performance of each feature set with the classification rate of the unseen test cases. Ten iterations of randomly selecting 60% of dataset 1 for training data and the remaining 40% for testing data are performed to evaluate the adequacy of the feature sets. Dataset 2 is also tested using the same training data to validate the results.

4.1.1 Feature Set A

Feature set A uses only the standard deviation of each channel within the sliding window. Table 1 summarizes the classification performance for data sets 1 and 2 using differing numbers of hidden states for the generated hidden Markov models. Tables 2 and 3 show typical confusion matrices for one of the 10 iterations when the number of hidden states is set at 6. The confusion matrices illustrate that the single feature of standard deviation per channel over a window is surprisingly good at recognition for just three features in total. The confusion matrix in Table 2 shows only minor errors in recognition. The confusion matrix of Table 3 however, shows a fairly significant error of 6 classification errors from 10 for the *One Short* gesture being the Four gesture. This error can somewhat be expected when the actual gesture movement is considered. Standard deviation by itself doesn't describe if acceleration increased or decreased, but merely the amount of variation that occurred.

	Classification Rate	
# States	**Test Set 1**	**Test Set 2**
N = 4	89.83	88.3
N = 5	90.28	92.3
N = 6	94.49	93.6
N = 7	93.14	89.6

Table 1. Recognition rates for Feature Set A

01	02	03	04	05	06	07	08	09	10	← classified as
17	0	0	0	0	0	0	0	0	0	01: Dead Ball
0	17	1	0	0	0	0	0	0	0	02: Four
0	0	14	2	0	1	0	0	1	0	03: Last Hour
0	0	0	18	0	0	0	0	0	0	04: Leg Bye
0	0	0	0	16	0	1	0	0	0	05: No Ball
0	0	0	0	0	18	0	0	0	0	06: One Short
0	0	0	0	0	0	17	0	0	1	07: Out
0	0	0	0	0	0	0	18	0	0	08: Penalty Runs
0	0	0	0	0	0	0	2	16	0	09: TV Replay
0	0	0	0	0	0	1	0	0	17	10: Wide

Table 2. Confusion matrix for 6 states, feature set A, dataset set 1

4.1.2 Feature Set B

Feature set B extends on feature set A by including standard deviation of each channel and the root mean square of the vector magnitude for each data window, Table 4 summarizes the classification performance for data sets 1 and 2 using differing numbers of hidden states for the Markov models generated. Tables 5 and 6 show typical confusion matrices for one of the 10 iterations when the number of hidden states is set at 7. The root mean square of the vector magnitude performs very similarly to feature set B, as expected, since the root mean square and average of the vector magnitude provide very similar information.

01	02	03	04	05	06	07	08	09	10	← classified as
10	0	0	0	0	0	0	0	0	0	01: Dead Ball
0	10	0	0	0	0	0	0	0	0	02: Four
0	0	9	1	0	0	0	0	0	0	03: Last Hour
0	0	0	10	0	0	0	0	0	0	04: Leg Bye
0	0	0	0	10	0	0	0	0	0	05: No Ball
0	**6**	0	0	0	4	0	0	0	0	06: One Short
0	0	0	0	0	0	10	0	0	0	07: Out
0	0	0	0	0	0	0	10	0	0	08: Penalty Runs
0	0	0	0	0	0	0	0	10	0	09: TV Replay
0	0	0	0	0	0	0	0	0	10	10: Wide

Table 3. Confusion matrix for 6 states and feature set A using test set 2

	Classification Rate	
# States	**Test Set 1**	**Test Set 2**
N = 4	94.94	99.2
N = 5	94.38	94.2
N = 6	97.19	98.4
N = 7	98.31	94.4

Table 4. Recognition rates for Feature Set B

01	02	03	04	05	06	07	08	09	10	← classified as
16	0	0	0	0	0	0	0	1	0	01: Dead Ball
0	18	0	0	0	0	0	0	0	0	02: Four
0	0	16	2	0	0	0	0	0	0	03: Last Hour
0	0	0	18	0	0	0	0	0	0	04: Leg Bye
0	0	0	0	15	0	0	0	0	2	05: No Ball
0	0	0	0	0	18	0	0	0	0	06: One Short
0	0	0	0	0	0	18	0	0	0	07: Out
0	0	0	0	0	0	3	15	0	0	08: Penalty Runs
0	0	1	0	0	0	0	1	16	0	09: TV Replay
0	0	0	0	0	0	0	0	0	18	10: Wide

Table 5. Confusion matrix for 7 states and feature set B using test set 1

4.1.3 Feature Set C

Feature set C also extends on feature set A by including standard deviation of each channel and the root mean square of each channel for each sliding window. Table 7 summarizes the classification performance for data sets 1 and 2 using differing numbers of hidden states for the Markov models generated. Tables 8 and 9 show typical confusion matrices for one of the 10 iterations when the number of hidden states is set at 4. This feature set combined with the number of HMM states is obviously adequate for the modeled gestures with both confusion matrices showing no confused test gestures. The key difference between this feature set and the other feature sets evaluated here is that in this feature set, the orientation of the arm is able to be established

01	02	03	04	05	06	07	08	09	10	← classified as
10	0	0	0	0	0	0	0	0	0	01: Dead Ball
0	10	0	0	0	0	0	0	0	0	02: Four
0	0	10	0	0	0	0	0	0	0	03: Last Hour
0	0	0	10	0	0	0	0	0	0	04: Leg Bye
0	0	0	0	10	0	0	0	0	0	05: No Ball
0	**7**	0	0	0	3	0	0	0	0	06: One Short
0	0	0	0	0	0	10	0	0	0	07: Out
0	0	0	0	0	0	0	10	0	0	08: Penalty Runs
0	0	0	0	0	0	0	0	10	0	09: TV Replay
0	0	0	0	0	0	0	0	0	10	10: Wide

Table 6. Confusion matrix for 7 states and feature set B using test set 2

since we use a magnitude of acceleration per channel.

	Classification Rate	
# States	**Test Set 1**	**Test Set 2**
N = 4	98.87	99
N = 5	97.86	96
N = 6	98.54	97.2
N = 7	99.43	94

Table 7. Recognition rates for Feature Set C

01	02	03	04	05	06	07	08	09	10	← classified as
17	0	0	0	0	0	0	0	0	0	01: Dead Ball
0	18	0	0	0	0	0	0	0	0	02: Four
0	0	18	0	0	0	0	0	0	0	03: Last Hour
0	0	0	18	0	0	0	0	0	0	04: Leg Bye
0	0	0	0	17	0	0	0	0	0	05: No Ball
0	0	0	0	0	18	0	0	0	0	06: One Short
0	0	0	0	0	0	18	0	0	0	07: Out
0	0	0	0	0	0	0	18	0	0	08: Penalty Runs
0	0	0	0	0	0	0	0	18	0	09: TV Replay
0	0	0	0	0	0	0	0	0	18	10: Wide

Table 8. Confusion matrix for 4 states and feature set C using test set 1

01	02	03	04	05	06	07	08	09	10	← classified as
10	0	0	0	0	0	0	0	0	0	01: Dead Ball
0	10	0	0	0	0	0	0	0	0	02: Four
0	0	10	0	0	0	0	0	0	0	03: Last Hour
0	0	0	10	0	0	0	0	0	0	04: Leg Bye
0	0	0	0	10	0	0	0	0	0	05: No Ball
0	0	0	0	0	10	0	0	0	0	06: One Short
0	0	0	0	0	0	10	0	0	0	07: Out
0	0	0	0	0	0	0	10	0	0	08: Penalty Runs
0	0	0	0	0	0	0	0	10	0	09: TV Replay
0	0	0	0	0	0	0	0	0	10	10: Wide

Table 9. Confusion matrix for 4 states and feature set C using test set 2

4.2. Continuous Gesture Recognition

Many sports gestures may require holding a limb in a certain position to allow other people in the scene to see the gesture. In the cricket gestures we model, four of them have this component. In the *No Ball* gesture for example, the arm is held out to the side of the body at shoulder height for players and other officials to see, then returned to the body. Figure 3 (b) shows a sequence of gestures where no movement exist as part of several gestures. Figure 3 (a) shows the corresponding log likelihood of the Gaussian stationarity model for the sequence. Since our approach for segmentation searches for periods of no movement, gestures which have this stationarity would be split in to multiple areas of movement, as is shown in Figure 3 (c). Obviously if we treated contiguous areas of movement as complete candidate gestures, we would falsely detect the subsequent parts of movement as new gestures. To alleviate this problem we introduce a minimum gesture length for segmentation. We assume that when movement occurs, a gesture is being executed, thus the movement must last for at least as long as the shortest gesture. Introducing a parameter such as this can lead to problems if it is not set to an appropriate value. Figure 4 shows the affect of setting the minimum gesture length to different sizes for the regions of movement identified in Figure 3. The candidate gesture areas are indicated by the dashed vertical lines with the arrows between. To demonstrate the performance of our segmentation method, we perform several sequences of umpire gestures and let the recognition system segment and classify the gestures. Gestures are not being performed continuously without a pause in between, rather there is a continuous stream of gestures sent to the segmentation system. Table 13 lists the results of the segmentation and classification of the test sequences. In sequence 1 for example, the table reads that of the 10 actual gestures in the sequence, the segmentation using a 2 second minimum gesture length correctly segmented 5 gestures; the 3.5 second minimum gesture length correctly segmented all 10 gestures; and the 5 second minimum gesture

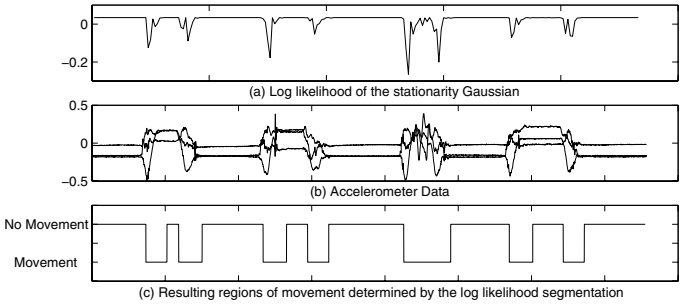

Figure 3. Partial sequence displaying areas of movement

	# Correctly Segmented			Classification Rate		
Sequence	2sec	3.5sec	5 sec	2sec	3.5sec	5 sec
1	5/10	10/10	2/10	10/12	10/10	4/6
2	6/10	10/10	3/10	13/14	10/10	6/7
3	4/10	10/10	3/10	7/12	10/10	4/7
4	3/10	10/10	3/10	8/14	10/10	5/8
5	6/10	10/10	2/10	9/14	10/10	4/7

Table 10. Recognition rates for Segmentation

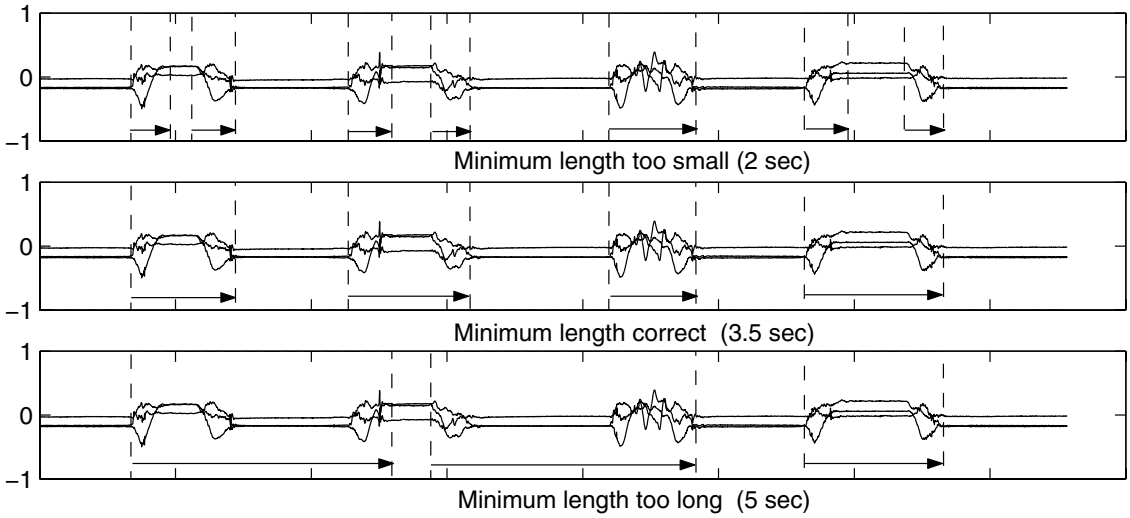

Figure 4. Differing values for the minimum gesture length parameter

length correctly segmented only 2 gestures. The table also reads that for sequence 1 the 2 second minimum gesture length correctly classified 10 of the 12 segments it detected; the 3.5 second minimum gesture length correctly classified all 10 segmented gestures; and the 5 second minimum gesture length correctly classified 4 of the 6 gestures it detected. The 5 and 2 second minimum gesture lengths have detected incorrect numbers of gestures since they combine two gestures into one or split one gesture into several, respectively, as illustrated in Figure 4.

The results of the table indicate that when the minimum length of a gesture is set to the correct value, the gesture recognition performs well. If the minimum length of a gesture is set too high or too low then the grouping of regions of movement in a sequence can degrade the recognition performance significantly. The true minimum value for the cricket gestures considered is approximately 3.5 seconds and corresponds to both the *Out* and *No Ball* gestures. The minimum gesture length parameter thus has to be tailored to the gestures and actor in question. In real games, however, the time between consecutive gestures is comparitively large, thus grouping of two adjacent gestures would be unlikely.

5. Conclusions and Future Work

We have presented results on the recognition of intentional human gesture for video annotation and retrieval. The novelty of our work is that we apply the recognition of gesture to automatic labeling of video. Actors who perform intentional gestures wear accelerometer sensors in the form of wrist bands. Gesture recognition is then performed in the sensor domain which avoids the problems associated with accurate image segmentation. We apply our approach to the recognition of sports umpire gestures using the hidden Markov model and a variety of feature sets. Results show that our recognition system is capable of recognising a set of 10 umpire gestures from the game of cricket and performs best when using a feature set . The work also shows the performance of segmenting gestures from a stream of continuous gestures by selecting candidate gestures by the existence of movement.

References

[1] ASG. Acceleration sensing glove. `http://bsac.eecs.berkeley.edu/~shollar/fingeracc/fingeracc.html`, 2000.

[2] A. Benbasat and J. Paradiso. An inertial measurement framework for gesture recognition and applications. In I. Wachsmuth and T. Sowa, editors, *Gesture Workshop*, pages 9–20. Springer-Verlag, 2002.

[3] Y. Gong, L. T. Sin, and C. Chuan. Automatic parsing of TV soccer programs. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 167–174, 1995.

[4] K. Hinckley, J. Pierce, M. Sinclair, and E. Horvitz. Sensing techniques for mobile interaction. *Symposium on User Interface Software and Technology, CHI Letters*, 2(2):91–100, 2000.

[5] H. Pan, P. V. Beek, and M. I. Sezan. Detection of slow-motion replay segments in sports video for highlights generation. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, 2001.

[6] T. Sensorglove. TUB Sensorglove. `http://www-mat.ee.tu-berlin/research/ac-sensor/ac-sens.htm`, 2000.

[7] W. Zhou, A. Vellaikal, and C. Kuo. Rule–based video classification system for basketball video indexing. In *ACM Multimedia 2000*, Los Angeles, USA, 2000.