An approach to automate the scorecard in cricket with computer vision and machine learning

Conference Paper · December 2017

DOI: 10.1109/EICT.2017.8275204

4 authors, including:

Md Asif Shahjalal
University of Michigan-Dearborn
6 PUBLICATIONS **53** CITATIONS
SEE PROFILE

Lamia Alam
Chittagong University of Engineering & Technology
15 PUBLICATIONS **96** CITATIONS
SEE PROFILE

2017 3rd International Conference on Electrical Information and Communication Technology (EICT), 7-9 December 2017, Khulna, Bangladesh

An Approach to Automate the Scorecard in Cricket with Computer Vision and Machine Learning

Md. Asif Shahjalal[1], Zubaer Ahmad[2], Rushrukh Rayan[3], Lamia Alam[4]
Department of Electrical and Electronic Engineering[1]
Department of Computer Science and Engineering [2 3 4]
Chittagong University of Engineering and Technology
Chittagong-4349, Bangladesh
asif.cuet93@gmail.com[1], mail2zubaer@gmail.com[2], rushrukhryan@gmail.com[3], lamiacse09@gmail.com[4]

***Abstract*—Cricket is beyond the shadow of a doubt one of the most popular forms of sports in the southern region of Asia. This form of sports is widely played in more than 125 countries recognized by the International Cricket Council. With the flourishing of cricket, various aspects of the game are being automated with the advent of technology. Use of computer vision in assisting third umpire decision is indubitably the well-liked one. One of the most challenging issues that first initiates the discussion on its prosperity is the duration of the game. The on-field umpire has to authorize decisions almost after each delivery following which modifications are carried out in the scorecard which is a very tedious process. The traditional approaches that have been considered so far involve wearing a specialized hand glove, which collides with the beauty of the originality in the field. In this paper, an approach is proposed and a prototype is implemented to automate the umpires decision by interpreting his hand gesture. The region of interest is selected using a Haar-cascade-classifier and then the particular gesture is recognized using logistic regression. This process would eliminate the manual updating of scorecards and thereby reduce the game duration notably. In addition, it excludes the prerequisite of wearing special gloves involving sensors. The efficiency of the algorithm is then cross-checked with the training and test data. This proved to be a very simple but efficient algorithm for umpires gesture detection.**

***Keywords*—Machine learning, scorecard automation, haar-cascade-classifier, computer vision, gesture detection, logistic regression.**

I. Introduction

Gestures are known as expressive, meaningful body movement which involves physical motion of the fingers, hands, arms, head, face, or body with the intent of: 1) assigning meaningful information or 2) interacting with the environment. In recent years gestures are widely used by humans to interact with computers and machines. Many common everyday equipment like TV, Smartphone, Car dashboard etc. can now be controlled by simple hand gestures. Gestures are also applied in many fields [1] like developing aids for the hearing impaired, enabling very young children to interact with computers, recognizing sign language, medically monitoring patients emotional states or stress levels, lie detection, monitoring automobile drivers alertness/ drowsiness levels, etc.

Involvement of gesture recognition technology in sports will make the gameplay fairer and proficient. The gestures performed by the sports officials which indicate to what is going on in the game. Which also can provide something meaningful about a player, a team, or the entire game. If the gestures of these officials are able to be recognized, meaningful information can be derived. We refer to a gesture as an intentional action by a person whereby part of the body is moved in a predefined way to indicate a specific event. Detecting these events enables automatic generation of highlights, contextual labeling of video and more importantly helps in decision making and automatic score update.

Cricket is the most popular sport in Bangladesh. It is the 2nd most famous sports in Asia, 4th in Europe and also 2nd most famous sport in the whole world [2]. But from the 19th century the same old manual method is being use to update the scoreboard, which is a great burden for the scorekeeper. The way of viewing the score has changed a lot in time, but the basic score updating process is still the same and performed by a person. So in the 21st century an automatic system is very much needed at this sector. Automatic systems are taking places of many boring manual task which was performed by humans 5 or 10 years ago. So developing a fully operational automated system like this is in dire need. Using modern equipments and machine learning algorithms we can increase the accuracy of the decision and provide flawless result what the naked eye misses. This motivates us to design a system which will recognize gesture of cricket umpire in real time. Besides the reason for choosing cricket is because it is the most popular sports in Bangladesh and unlike other popular sports it doesnt have enough technological application to make it more fair and accurate.

Gesture recognition has been explored in both vision and sensor based. In vision based system some image processing or computer vision based method is used to recognize the gesture. On the other hand, in sensor based system the person

978-1-5386-2307-7/17/$31.00

who perform the gesture wears sensor/sensors, and when the gesture is performed the sensor data taken and used to identify the gesture. In the area of sports and other sector, several attempts have been made for gesture recognition using both sensor-based and vision based technique.

Sensor-based hand gesture recognition started with the invention of glove-based system, which was further divided into two distinct categories- active data glove and passive data glove over the years [3]. The first glove-based systems were designed as a part of a camera-based LED system to track body and limb position for real-time computer graphics animation in the 1970s, and since then, a number of different designs have been proposed [4]. A low-cost version named the Power Glove [5], was commercialized by Mattel Intellivision as a control device for the Nintendo video game console in 1989 and became well known among video games players. Chambers et. al proposed a probabilistic hierarchical framework to extract gestures and significant events of Kung Fu martial art movements acted out by an instructor in a simulated training video using accelerometers and the Hidden Markov Model [6]. In another work, Chambers et. al proposed the use of the Hierarchical Hidden Markov Model (HHMM) in conjunction with a filler model for segmenting and classifying gestures at differing levels of detail. In this work, sports video was augmented with accelerometer data from wrist band worn by umpires in the game. The gestures they recognized are: Dead Ball, Four, Last Hour, Leg-Bye, No Ball, One Short, Out, Penalty Runs, TV Replay and Wide [7], [8]. Another popular waay to recognize hand gesture is to use kineect [9].

Gesture detection was successfully recognized using a feature vector and a real-time histogram based algorithm by Gomez et. al.[10]. Wang et. al recently developed a simple and inexpensive for 3D articulated user-input using the hands [11]. Their approach uses a single camera to track a hand, wearing an ordinary cloth glove that is imprinted with a custom pattern. Bhansali et. al proposed a method of Gesture recognition to Make Umpire Decisions by using subtraction method and gradient method [12]. They tested the subsequent six gestures particularly OUT, SIX, NEWBALL, NO-BALL, DEADBALL and FOUR. Google is recently developing a system called project soli [13] where they use special radar sensor to detect the fingers movement. Soli sensor technology works by emitting electromagnetic waves in a broad beam. Objects within the beam scatter this energy, reflecting some portion back towards the radar antenna. Properties of the reflected signal, such as energy, time delay, and frequency shift capture rich information about the objects characteristics and dynamics, including size, shape, orientation, material, distance, and velocity. In project soli there are some sensors are used but those are not attached with the gesture performers hand or body. So these can be called as a hybrid system where sensor and radar vision both are used simultaneously.

Sensor-based systems had limited accuracy and were tethered to computers using cumbersome wiring [4-8]. They were meant for very specific applications and were never commercialized. Sensor-based techniques, however, have advantage that they can be used in much less constrained domains and are not reliant on lighting conditions or camera calibration. But vision based gesture recognition has to concentrate on recognition of individual gestures in constrained environments. Most of the vision based works stated above are designed only for detecting static hand gesture [9, 10]. They were unable to detect the dynamic hand gesture in constrained environment.

In contrast to these, we intended to design a vision-based system using logistic regression machine learning technique which can detect both static and dynamic hand gestures of a cricket umpire. Our goal is trying to recognize the gestures an umpire performs in a match and update the scoreboard for the corresponding gesture accurately. We trained a haar-cascade-classifier to detect human wrists from the video stream from a static camera. Then the selected region of interest is continuously checked through the multiclass logistic regression model if a gesture is matched. Then accordingly the score is updated.

II. Proposed Methodology

Our proposed method for recognizing cricket umpire gesture from real time video using logistic regression algorithm has basic two parts. These are: 1) Learning Phase, 2) Recognition phase. In the learning phase we have to give a lot of gesture images as training data for input. Each of the training image is resized to 20x20 pixel. This is done for fast and efficient computation. Then the resized images are converted to grayscale which is also for efficiency. Then all the images put into a matrix where each row of the matrix refers the pixel intensity of an image. The last column of the matrix represents the output (which gesture picture is in the corresponding image) of that image. Then using the matrix cost function is calculated and minimized by using gradient descent algorithm. For the lowest value of the cost function the corresponding parameter values are consider as weight .

Now on the recognition phase we capture video from a static camera or any other source, then we separate the frames of the video. After separating to select the region of interest from the input images we use Haar-Cascade-Classifier. After selecting the region of interest we convert it to a 20x20 pixel image , vectorize it and send the image feature value X and previously obtain weight to logistic regression hypothesis. The hypothesis gives the probability of a gesture to be true or false. The recognition step works in real time environment and it can provide output at a very decent amount of time.

A. Learning Phase

1) Input Training Image: In first phase of learning the we have to collect a lot of sample data. We use cricket umpires gesture RGB image as input data. It is better to collect a large amount and accurate data because the performance of our system is directly proportional to the input data. Usually all the decisions in a cricket match involve some kind of gesture by hand from the umpire. So we particularly crop the wrist portion of the image which fairly defines a decision. After

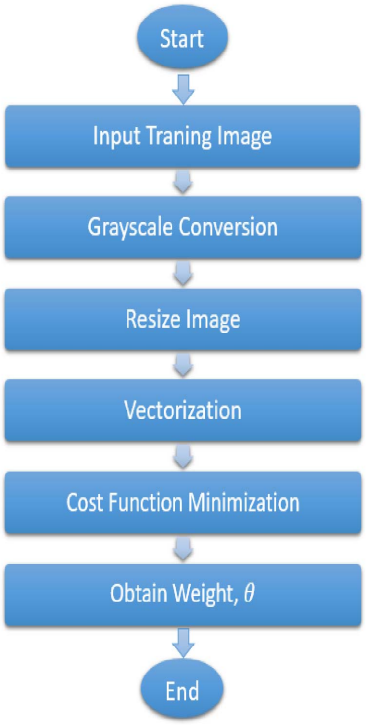


Fig. 1: Proposed Framework for Learning

collecting data for each gesture we have to give them as an input of the system.

2) Grayscale Conversion: Now we convert all the given RGB training images to grayscale. A grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest. Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. But also they can be synthesized from a full color image.

3) Resize Image: After giving all the training images we have to resize them to a very smaller version. This is because the image we give input are very high resolution which need much power and time to compute. So to increase efficiency we convert all the sample image into a smaller size. We resize the image into 20x20 pixel, which is very small in size, but sufficient for calculation.

4) Vectorization: In logistic regression we need to find out the feature value of each training image. So we take the pixel intensity of each pixel and create a two dimensional matrix. In our case we iterate through the previously resized 20x20 pixel images and save each image pixel values in each row of the matrix. Each column in the matrix represents a pixel value and each row represents a single image. Each row of the matrix contains 401 values where first 400 values are the pixel intensity of the image and the last value is the output of that image. All images from different gesture are stored in the same matrix.

5) Cost Function Minimization: We know the hypothesis of logistic regression is:

$$P(y=1|x) = h_\theta(x) = \frac{1}{1+\exp(-\theta^T x)} \equiv \sigma(\theta^T x) \quad (1)$$

$$P(y=0|x) = 1 - P(y=1|x) = 1 - h_\theta(x) \quad (2)$$

And the cost function is:

$$J(\theta) = -\sum_i \left(y^i \log(h_\theta x^i) + (1-y^i)\, \log(1-h_\theta x^i) \right) \quad (3)$$

We now have a cost function that measures how well a given hypothesis h_θ fits our training data. We have to minimize the cost function to find out the optimal value of the weight θ. So we have our hypothesis function and we have a way of measuring how well it fits into the data. Now we need to estimate the parameters in the hypothesis function. That's where gradient descent is used.

Suppose that we graph our hypothesis function based on its fields θ_0 and θ_1.We put θ_0 and θ_1 on the x and y axis and the cost function on the vertical z axis respectively. The points on the graph will be the result of the cost function using the hypothesis with those specific θ parameters. We will have succeeded when our cost function is at the very bottom of the pits in our graph, i.e. when its value is the minimum.

6) Obtain Weight(θ): After we get the minimum value of J(θ) we take the corresponding values of θ which is our desired weight. We save the values of θ in a matrix and our learning phase is complete.

B. Recognition Phase

1) Video Input: The first step of recognition phase is gesture video input. User have to open the program and perform an umpire gesture in front of the camera. This video is taken as input to the system and send for further analysis.

2) Frame Separation: When we have the input video, we have to separate the frame to analyze them. We will use each frame as an image. We use vectorization as before to keep the image as a matrix. Frame separation is applied because it is quiet hard to analyze the whole video rather than analyze a frame. So in video processing it is common use to separate the frames and then analyze those as an individual image.

3) Region of Interest Selection: Our goal is to recognize the gesture from an image, therefore we use only gesture images to train our system. But the image we got as an input there can be other things too. So we have to find the gesture part from the image which is our region of interest. In order to do that we use Haar-Cascade classifier algorithm. Haar-Cascade classifier can be used to classify a specific type of object. In our case we design the classifier to recognize human hand because most of the gesture of cricket umpire is performed by

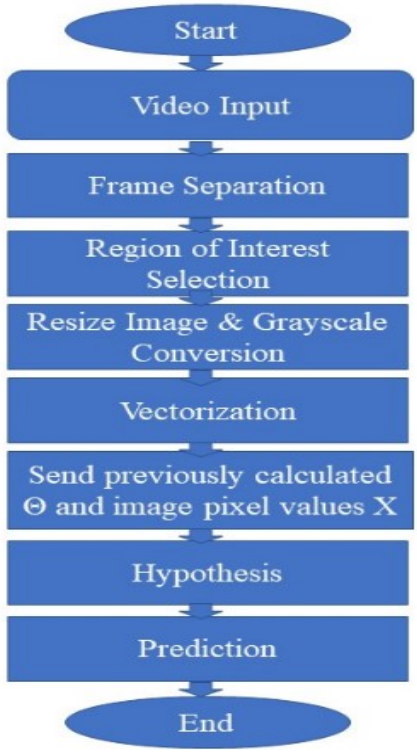


Fig. 2: Proposed Framework for Recognition

hand. After recognizing the hand area from the input image it crops that specific part of the image and we do our further calculation for that part only.

4) Resize Image and Grayscale Conversion: After selecting the region of interest, they are converted to grayscale format and resized into 20x20 pixel. This is because a high resolution image will take much longer time and computational power to compute, which is not preferable for real time operation.

5) Prediction from Hypothesis: In logistic regression it output a binary result "yes" or "no" which is derived as 1 or 0. So we use a hypothesis class to try to predict the probability that a given example belongs to the "1" class versus the "0" class. Specifically, we will try to learn a function of the form:

$$P(y=1|x) = h_\theta(x) = \frac{1}{1+\exp(-\theta^T x)} \equiv \sigma(\theta^T x) \quad (4)$$

$$P(y=0|x) = 1 - P(y=1|x) = 1 - h_\theta(x) \quad (5)$$

To get a prediction from the hypothesis function we need to calculate feature value X from the input image and weight θ from the training image. The form for our hypotheses $h_\theta(x)$ to satisfy $0 \leq h_\theta(x) \leq 1$. This is accomplished by plugging $\theta^T x$ into the Logistic Function.

$$h_\theta(x) = g(\theta^T x) \quad (6)$$

$$z = \theta^T x \quad (7)$$

$$g(z) = \frac{1}{1+e^{-z}} \quad (8)$$

The function g(z), shown here, maps any real number to the (0, 1) interval, making it useful for transforming an arbitrary-valued function into a function better suited for classification. $h_\theta(x)$ will give us the probability that our output is 1. For example, $h_\theta(x)$=0.7 gives us a probability of 70% that our output is 1. Our probability that our prediction is 0 is just the complement of our probability that it is 1 (e.g. if probability that it is 1 is 70%, then the probability that it is 0 is 30%). We used a multiclass one-vs-all classifier where each gesture is trained individually by choosing all other gestures to be not true.

III. Result

For the experiment 5 different gestures were taken. 1000 samples were taken for each gesture. The samples were preprocessed. The images were converted to greyscale and resized into 20/20 pixel images. The pixel values were saved in a matrix. The last column of each row defined which gesture does it represent. This way a one-vs-all logistic regression model was used to build logistic regression classifier.

A few sample images are shown in Fig. 3.

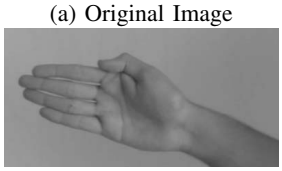
Fig. 3: Sample input gestures for four (on top), six (on left), wide-ball (on middle) , no-ball (on bottom right), out (bottom middle)

The sample images are then converted to greyscale as in Fig. 4.

The region of interest is selected by the haar-cascade-classifier and then the logistic regression model predicts from the resized region of interest which gesture matches most with it. According to the matching the score is updated. A few output from the experiment are shared in Fig. 5.

A statistics of the performance is given below :

(a) Original Image



(b) Greyscale Image

Fig. 4: Greyscale conversion

We trained our system to recognize five different gestures. These gestures are delivered by 10 different users. Each of them deliver these 5 gesture multiple times and the success rate are calculated depending on accurately detection of those gestures by the system. We take those data and measure the accuracy of our system in two different ways 1) Accuracy rate based on different people and 2) Accuracy rate based on Different Gesture Accuracy rate based on different people: In this section the sample gesture along with its successful rate for each different person are discussed. For each different person the successful rate can be mentioned as following theory: SR= GR/GD*100Here, SR=Success rate for each different person GR=Total number of gesture successfully recognized by the system GD= Total number of gesture delivered by the user. Following the above theory, the chart with successful rate is found given in following.

TABLE I: Success rate of different people

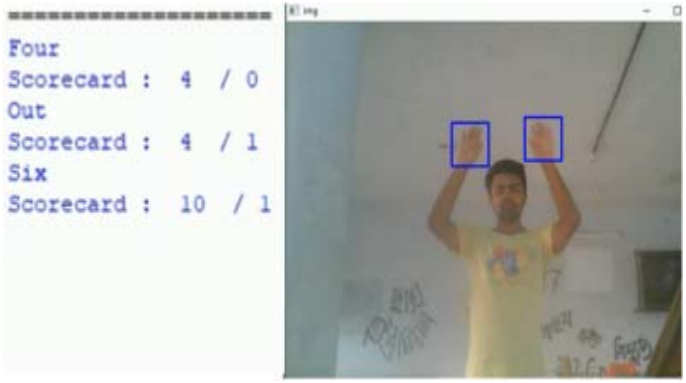
Person	GD	GR	SR
1	75	68	90.66
2	75	69	92.00
3	75	66	88.00
4	75	70	93.33
5	75	67	89.33
6	75	65	86.66
7	75	71	94.66
8	75	69	92.00
9	75	68	90.66
10	75	70	93.33

Now if the average successful rate is considered over this chart than mean value will be 91.06%.

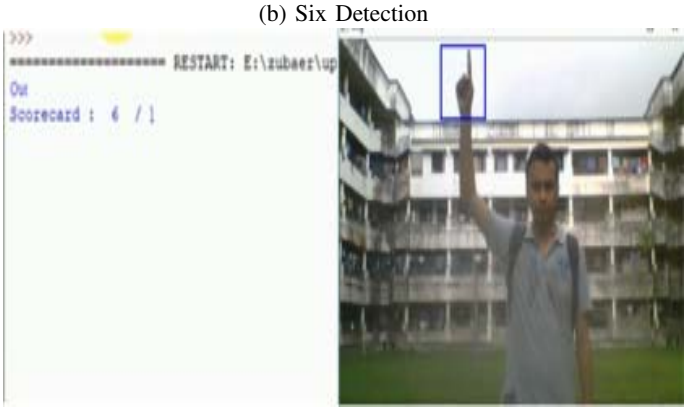
Accuracy Rate of Different Gesture: Previously result for different person are given. Now in cricket gesture for different event are different. Some are simple and some are complex. Some required the use of two hands and some required use of only one hand. Time is also variant for different gesture. Though much complex gesture is neglected. Here the accuracy rate for each gestures are given.

Now if the average successful rate is considered over this chart than mean value will be (88.00+90.00+86.00+83.33+93.33)/5 = 88.13 A total of 10 people participated to test our system, each of different

(a) No Ball Detection



(b) Six Detection



(c) Out Detection

Fig. 5: Scoreboard Update

TABLE II: Accuracy rate for different gesture

Gesture	GD	GR	SR
Out	150	132	88.00
No-Ball	150	135	90.00
Wide-Ball	150	129	86.00
Four	150	125	83.33
Six	150	140	93.33

body structure. We trained our system to recognize five cricket umpires gesture. Depending on these two criteria success rate is found which already being mentioned. If the results are shown in graph chart than we get the following charts.

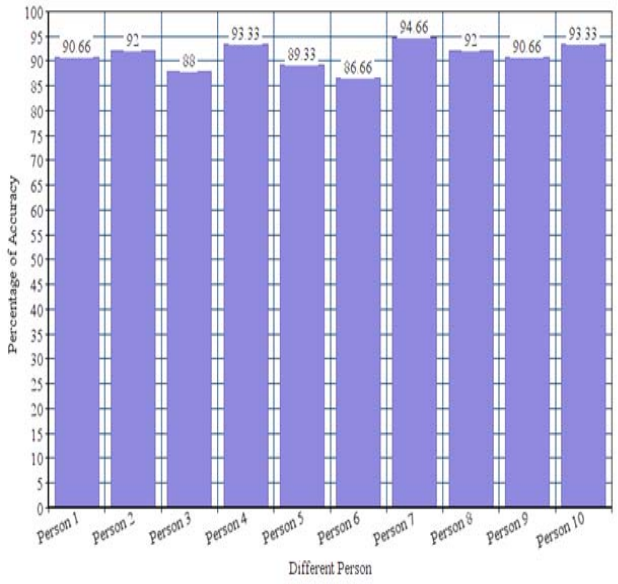


Fig. 6: Success rate depending on person

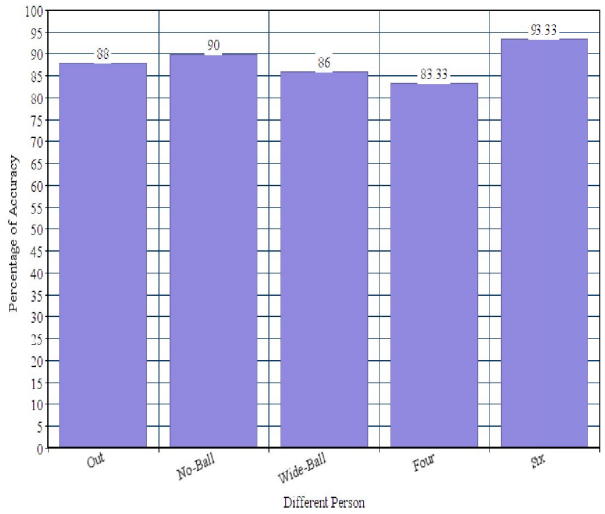


Fig. 7: Success rate depending on gesture

IV. Conclusion

The novelty of our work is that it successfully detects the appropriate gestures in the game of cricket without the help of any specialized sensor or gloves. Only a static camera is needed on the umpire. As the background is mostly fixed the background subtraction is very easily done. However this approach worked better for static gestures like out, no, six, wide etc. However in case of dynamic gesture like four the efficiency was found to be poor. There is a room of improvement in this case.

Multiple classifiers were need to be trained in order to make it work. However, a much easier and simple solution to find out the region of interest can be found out which will decrease the complexity of the program as well as increase the efficiency with respect to time.

References

[1] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[2] "Top 10 list of the internet world's most popular sports," http://www.topendsports.com/world/lists/popular-sport/fans.htm.

[3] P. Premaratne, "Historical development of hand gesture recognition," in *Human Computer Interaction Using Hand Gestures*. Springer, 2014, pp. 5–29.

[4] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer graphics and Applications*, vol. 14, no. 1, pp. 30–39, 1994.

[5] D. L. Gardner, "The power glove," *Des. News*, vol. 45, pp. 63–68, 1989.

[6] G. S. Chambers, S. Venkatesh, G. A. West, and H. H. Bui, "Hierarchical recognition of intentional human gestures for sports video annotation," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2. IEEE, 2002, pp. 1082–1085.

[7] G. S. Chambers, S. Venkatesh, and G. A. West, "Automatic labeling of sports video using umpire gesture recognition," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2004, pp. 859–867.

[8] G. S. Chambers, S. Venkatesh, G. A. West, and H. H. Bui, "Segmentation of intentional human gestures for sports video annotation," in *Multimedia Modelling Conference, 2004. Proceedings. 10th International*. IEEE, 2004, pp. 124–129.

[9] Y. Li, "Hand gesture recognition using kinect," in *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*. IEEE, 2012, pp. 196–199.

[10] I. Gómez-Conde, D. Olivieri, X. A. Vila, and S. Orozco-Ochoa, "Simple human gesture detection and recognition using a feature vector and a real-time histogram based algorithm," *Journal of Signal and Information Processing*, vol. 2, no. 04, p. 279, 2011.

[11] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," in *ACM transactions on graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 63.

[12] L. Bhansali and M. Narvekar, "Gesture recognition to make umpire decisions," *International Journal of Computer Applications*, vol. 148, no. 14, 2016.

[13] "Your hands are the only interface you'll need," https://atap.google.com /soli/.