International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: 6.614
Available Online at www.journalijcar.org
Volume 7; Issue 5(A); May 2018; Page No. 12248-12257
DOI: http://dx.doi.org/10.24327/ijcar.2018.12257.2146

Research Article

REAL-TIME WHOLE-BODY ACTION RECOGNITION IN VIDEOS USING THRESHOLD HIDDEN MARKOV MODEL

Akinyokun O. C and Akintola K. G

Department of Computer Science, Federal University of Technology, Akure, Nigeria

ARTICLE INFO

Article History:

Received 5th February, 2018
Received in revised form 20th March, 2018 Accepted 8th April, 2018
Published online 28th May, 2018

Key words:

Threshold Hidden Markov Model, Silhouettes, Radial-signal distance features, Viterbi decoding, code-words.

ABSTRACT

In surveillance scenario, some actions by human beings can generate alert actions. For example, a person jumping up and down in some environments may trigger alert action. There are some studies on hand gestures and sign language recognition. Automatic recognition of whole-body gestures is required in surveillance environment. This is a complex task in terms of segmenting meaningful gesture patterns from whole-body features. Few papers have been able to do this but the features used during computation is time consuming and may not be adequate in real time surveillance environments which require timely reporting of human actions. In this paper, effort is made to use light-weight features which are easy to extract and compute for whole body gesture action recognition in videos. To extract these features, silhouettes of actors are extracted from videos using background subtraction method. The features termed radial-signal distance features are then extracted from these silhouettes to form the feature vectors. The features are then quantized to obtain the code-words. A Left-Right Hidden-Markov-Model (LRHMM) is then constructed for each meaningful action. A threshold model is also constructed from the concatenation of all the states of the key actions Hidden-Markov-Model (HMM) models. The Forward algorithm and Viterbi decoding are then employed to spot and recognise actions patterns using the constructed models. Experiment performed on some video actions using the radial signal distance features shows a recognition accuracy of 93.16%.

INTRODUCTION

nature, for example, walking, running, jumping, bending and so on. In this paper, action recognition made by the whole human body is considered. Action spotting is the technique of extracting meaningful actions from continuous input signals and recognizing them. This has found many areas of applications such as automatic control of appliances, human-computer interaction, intelligent surveillance and human-robot interaction. Recently, the computer vision community has been carrying out research on how actions can be recognized from videos. There are model-based approaches which employ a kinematics model to represent the poses of body parts in each snapshot of body action. The recognition algorithm first aligns the kinematic model to the observed body appearance in each video frame and then codes the motion of the body parts with the model transformations. There are the holistic approaches otherwise known as the appearance-based methods which make use of the appearance properties of each action frames without explicitly representing the kinematics of the human body (Bobick and Davis, 2001; Gonzalez *et al.*, 2001). There is the part-based method in which the appearance of an actor is deposed into a set of small local spatio-temporal components and statistical models are applied to map the local components to actions (Chen *et al.*, 2008).

It has been recognized that action spotting has two major difficulties, namely: segmentation and spatio-temporal variances. The segmentation aspect determines the start point and the endpoint of an action. As the performer of the action switches from one action to another, the body passes through many intermediate positions located between the two actions. Another challenge in action recognition is that the same action varies in shape and duration depending on the actors. Therefore, the action recognition algorithm needs to solve the problem of spatial and temporal variances simultaneously. In that work, Hidden Markov Model (HMM) is used for modeling the action spotting network because it can solve the problems highlighted above. It has been the most successful and widely used approach to model events which have spatio-temporal variances (Lee and Kim, 1999); Elmezain *et al.*, 2009). HMM is used to estimate the probability of the similarity of an input pattern with a reference pattern. The matching process of the HMM does not require additional consideration for reference patterns with spatial and temporal variances because they are internally represented as probabilities of each state and transition. In addition, the set of unknown patterns can be modeled using a threshold model. In

Corresponding author:* **Akinyokun O. C
Department of Computer Science, Federal University of Technology, Akure, Nigeria

building the threshold model however, it is not easy to train the garbage model that can best match non-action, that is, unknown patterns because the set of non-action patterns is not finite. To overcome this problem, internal segmentation property of the HMM is adopted and a threshold model that consists of states in trained action models and helps to determine the reliability of the matching results of action models is used.

Many real times HMM recognitions suffer from time delay. The algorithm that is proposed in this work uses a forward gesture spotting that recognizes gesture spotting and recognition simultaneously. It is recognized that the nature of features used for action recognition can be computationally costly and result into time delay. Several features have been adopted. Such include the view invariant moment, the angle from the vertical axis of somebody joints (Yang *et al.*, 2007), Binary local motion descriptors (Chen *et al.*, 2008) and Motion of Scale Invariant Feature Transform (MOSIFT) (Chen and Hauptmann 2008). Extracting these features is computationally expensive in terms of time. In this paper, the radial signal distance features are extracted from the appearance of each of the postures representing an action to form the features. These features are computationally less expensive than the previous features.

Review of Some Related Works in Action Recognition in Videos

Various methods have been adopted in gesture action recognition in videos. The holistic methods do not require the localization of body parts. Instead, global body structure and dynamics are used to represent human actions. The key idea is that, given a region of interest centered on the human body, global dynamics are discriminative enough to characterize human actions. Compared to approaches that explicitly use a kinematic model or information about body parts, holistic representations are much simpler since they only model global motion shapes (Klaser, 2010). In Bobick and Davis (2001), a holistic approach using temporal template for action recognition is presented. The work is motivated on the use of template matching for action recognition. The objective of the project is to construct a view-specific representation of actions, where an action is defined as motion over time. It is assumed that either the background is static or that the motion of the object can be separated from either camera-induced motion. A binary Motion Energy Image (MEI) and Motion History Image (MHI) are used to interpret human activity in an image sequence. First, a set of images in a sequence is extracted by using frame differencing and the set is accumulated over time to get the MEI. Then, the MEI was changed into MHI which is a scalar-valued image. Finally, these view-specific templates were matched against the stored models of views of known actions during the recognition process. The distance between the gallery features and probe features are calculated and the one with minimum value recognizes the action that the test image sequence denote. The approach is view dependent.

Online recognition of human activity for video surveillance is proposed in Gonzalez *et al.* (2001). The use of appearance based methods to recognize activities particularly using sequence of key frames motivate the development of the system. Activity sequence is described by a set of key frames that best represent the activity. A background subtraction method is used to extract the foreground. A skeletal representation is used to represent the Silhouette of an actor. Using different samples of activity sequences, an eigenspace is used to represent an activity as a set of points, each one corresponding to a frame of the sequences. Recognition is performed by comparing this skeleton with each first key frame and select the most similar, the consequent frames is compared with the next and so on. The limitation of the system is that effective ordering of the key frames is highly needed.

It has been recognized in recent times that local space-time features capture characteristic shape and motion information for a local region in video. They provide a relatively independent representation of events with respect to their spatio-temporal shifts and scales as well as background clutter and multiple motions in the scene. Such features are usually extracted directly from video and therefore avoid possible failures of other pre-processing methods such as motion segmentation or human detection (Klaser 2010). Feature detectors usually select characteristic spatio-temporal locations and scales in videos by maximizing specific saliency functions. In Laptev (2005), a feature detector based on a spatio-temporal extension of the Harris cornerness criterion is proposed. The cornerness criterion is based on the eigen-values of a spatio-temporal second-moment matrix at each video point. Local maxima indicate points of interest. The authors note the importance of using separate spatial and temporal scale values since spatial and temporal extent of events are in general independent. It is argued in Dollars *et al* (2005) that in certain cases, true spatio-temporal corner points used are relatively rare, while enough characteristic motion is still present. Therefore, they design their interest point detector to yield denser coverage in videos using spatial Gaussian kernels and temporal Gabor filters. The features are computationally expensive.

In Chen and Hauptmann (2008), a framework for recognizing human actions in surveillance videos using Motion of Scale Invariant Feature Transform (MOSIFT) is proposed. It is recognized that Local Spatio-temporal features around interest points provides compact but descriptive representations for video analysis. A MOSIFT algorithm is proposed which detects interest points and encodes not only their local appearance but also explicitly models motion unlike current approaches which implicitly model motion. A bigram model is introduced to construct a correlation between local features to capture the global structure of actions. This is a part-based method which involves detection of interest points, constructing a feature descriptor and building a classifier, hence the following is carried out in the research.

- MOSIFT Interest Point Detection: The detection of interest points is achieved using SIFT algorithm, then the detection of spatio-temporal features is done using motion constraints which consist of a sufficient amount of optical flow around the distractive points.
- MOSIFT Feature Description: A single feature descriptor, which concanates both Histogram of Gradient (HOG) and Histogram of Optical Flow (HOF) into one vector which is called early fusion is used as feature descriptor.
- MOSIFT Feature Classification: Support Vector Machine (SVM) is used as a classifier to recognize the action performed.

- The limitation of this work is that it is computationally expensive. Real time operations can be achieved only if specialized hardware is used. (Chen and Hauptmann, 2008).

In Chen *et al* (2008), an automatic algorithm capable of recognizing aggressive behaviours from videos using local binary motion descriptors is proposed. The proposed interest point detector is based on Harris Corner Detector. The points along edges with velocity vectors using second moment gradient matrix of gradient magnitudes of x, y and t are used. The goal is to find high contrast points both in space and time. This will identify the points which are along edges in a video image and contain velocity vectors. The formula for interest point calculation is as follows:

$$L(x,y,t,\sum) = I(x,y,t)^{*}\ g(0,\sum) \qquad (1)$$

$$R(x,y,t) = \sqrt{\left(\frac{\delta l}{\delta x}\right)^2 \left(\frac{\delta l}{\delta y}\right)^2 \left(\frac{\delta l}{\delta t}\right)^2} \qquad (2)$$

where *L* denotes a smoothed video which is computed by a convolution between the original video I, and a Gaussian smoothing kernel *g*. The features are computationally expensive.

Hidden Markov models have been used for activity recognition in videos. In Bashir *et al.* (2005), activity classification and recognition based on the trajectory traversed by an object is proposed. It is noted that motion trajectories provide rich spatiotemporal information about object's activity. The work presents novel classification algorithms for recognizing objects' activity using object motion trajectory. In the proposed classification system, trajectories are segmented at points of change in curvature and the Principal Component Analysis (PCA) is applied to change the data into sub-space representation. The objective of the work is to represent trajectories using compact and robust representation to capture the spatio-temporal movement patterns. It semantically describes meaningful high level descriptions of the activities, actions and events based on this trajectory data. A trajectory in that work is a 2-D N-tuple corresponding to the x and y axes projections of the object's centroid location at each instant of time. The trajectory is changed to sub-space representation using PCA. The coefficient of PCA is modeled using Gaussian Mixture Model (GMM). Once the training phase has been completed, each new trajectory is categorized as one of the learned classes of object motion based on the Maximum A Posteriori estimation (MAP). The limitation is that not all action can be represented by the trajectories of the actors.

In Yang *et al.* (2007), gesture spotting and recognition for Human Robot Interaction (HRI) is proposed. Previous HRI research focused on issues such as hand gestures, sign language and command gesture recognition. However, that research work proposes automatic recognition of whole-body gestures for HRI. It is recognized that modeling meaningful gesture patterns from whole-body gestures is a complex task. The work presents a new method for recognition of whole-body key gestures for HRI. A human subject is first described by a set of features, encoding the angular relationship between a dozen body parts in 3-D. A feature vector is then mapped to a codeword of hidden Markov models. In order to spot key gestures accurately, a transition gesture model is proposed. To reduce the states of the transition gesture model, model reduction which merges similar states based on data-dependent statistics and relative entropy is used. The experimental results demonstrate that the proposed method can be efficient and effective in HRI, for automatic recognition of whole-body key gestures from motion sequences. The features are computationally expensive.

System Description

Figure 1 shows the framework of the HMM-based system for gesture action recognition. It consists of the object detection module object tracking module, feature extraction, vector quantization, database of action codes, HMM models and action recognition modules.

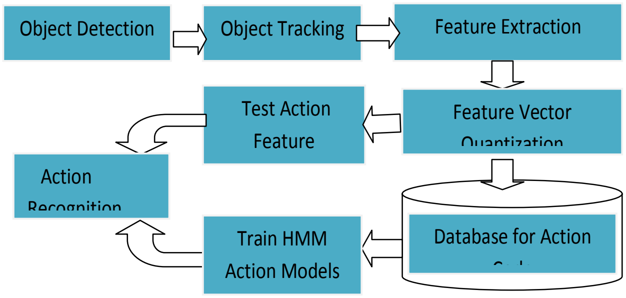

Figure 1 Proposed Gesture Action Recognition Flow Diagram

Object Detection

Kernel Density Estimation (KDE) is the mostly used and studied nonparametric density estimation algorithm. The model is the reference dataset, containing the reference points indexed natural numbered and has been used in (Akintola *et al.*, 2016; Akintola and Tavakkolie 2011; Elgammal *et al.*, 2002) for foreground detection. The algorithm assumed that a local kernel function is centered upon each reference point and its scale parameter (the bandwidth). The common choices for kernels include the Gaussian and the Epanechnikov kernel. The algorithm is presented as follows. Let $x_{1},x_{2},\ldots,x_{n,} \in R^{d}$ be a random sample taken from a continuous, univariate density *f*, KDE is given by:

$$\hat{f}(x,h) = \frac{1}{nh}\sum_{i=1}^{n} k\left(\frac{x-x_i}{h}\right) \qquad (3)$$

k(.) is the function satisfying:

$$\int k(x)dx = 1 \qquad (4)$$

k(.) is refered to as the Kernel, *h* is a positive number, usually called the bandwidth or window width.
The Gaussian Kernel is given by:

$$K_N = (2\pi)^{-\frac{d}{2}} \exp\left(-\tfrac{1}{2}r\right) \qquad (5)$$

where $r = ||x||^2$

Object Tracking

The proposed object tracking algorithm has been used in Akintola (2014). The algorithm is composed of two stages. First is the appearance correspondence mechanism. Once objects are detected, the appearance models are generated for objects appearing in the scene. The model is the estimate of probability distribution of pixel colours. Multiple models are developed for a single object, which are used in subsequent frames to match the set of currently detected models and that of target models. In the second phase, occlusion and object merge and separation are handled.

After foreground object have been detected, a bounding box is used to get the area occupied by the object. To cater for the spatial distribution of color and to increase the ability of the tracker the bounding box is divided into two regions, called the lower and the upper regions. The upper-region histogram and the lower region histogram are kept in a list referred to as the RO-Li. The currently detected objects in the scene are also kept in a list called CO-Li. The idea is to search the elements stored in CO-Li in RO-Li. This algorithm will be very fast since searching is done in the list and not in spatial window. For each object, the computed histograms is compared to that of reference histograms using the Bhattacharyya distance measure. Bhattacharyya Distance (BD) measure returns a range of bounded values in the range [0, 1]. '0' indicates that the two objects are very similar while '1' indicates they are not similar. If BD(obj1,obj2) < 0.419, the two objects can be regarded as still similar.

$$\text{CO-L}_i(i) = \underset{k}{\text{arg Min}}\, [D(\text{CO-L}_i , \text{RO-L}_{i-1}(k)], \text{ vk,vi} \tag{6}$$

Given a state vector $x_{t,} = (d_t,e_i) = (dx_t, dy_t, ex_t, ey_t)$, the candidate region where colour information will be gathered is defined as $R(x_{t,}) = d_t + e_tW$. Within this region, a kernel $q_t\,(x) = \{q_t(n;x)\}, n = 1, \dots, N\}$ density estimate of the colour distribution at time *t* is given by:

$$q_t(n;x) = K \sum_{u \epsilon R(x)} \delta[b_t(u) - n] \tag{7}$$

where δ is the Kronecker delta function, K is a normalization constant ensuring $\sum_{n=1}^{N} q_t(n;x) = 1$, and locations *u* lie on the pixel grid. This model associates a probability to each of the N states. At time *t,* the color model $q_t(x)$ will be compared to the reference color model $q^*(x) = \{q^*(n)\}, n = 1, \dots, N\}$ with $\sum_{n=1}^{N} q_t(n;x) = 1$. In this paper, the reference distribution gathered at an initial time t_0 at a location/scale x^*_{t0} automatically provided by a foreground detection module given as:

$$q^* = q_{t0}(x^*_{t0}) \tag{8}$$

The data likelihood must favor candidate color histograms close to the reference histogram. A distance measure therefore is needed to be chosen on the HSV color distributions. Such a distance is used in the deterministic techniques as the criterion to be minimized at each time step (Comanuciu et. al., 2000) In (Perez *et al.*, 2002; Comanuciu et. al., 2000), D is derived from the Bhattacharyya similarity coefficient, and is defined as:

$$\mathrm{D}(q^*, q_t(x))] = \left[1 - \sum_{n=1}^{N} \sqrt{q^*(n) q_t(n;x)}\right]^{\frac{1}{2}} \tag{9}$$

where distance between probability distributions is bounded within [0,1].

Feature Extraction

Radial signal distance features are directly calculated from the detected object's silhouettes. These features capture the shape and motion information of the activity (Akintola *et al.*, 2016). The centroid of the contour (c_x, c_y) is calculated using Equation (10). From the centroid, a pre-defined number of axes are projected outwards at specified regular angles to the nearest edges of the contour in an anti- clockwise direction as shown in Figures 2.

$$(c_x, c_y) = \frac{1}{n} \left(\sum_{t=1}^{n} x_t , \sum_{t=1}^{n} y_t\right) \tag{10}$$

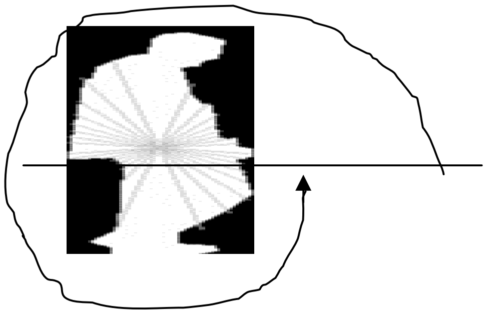

Figure 2 Radial Distance Shape Features Extraction

The distance of a line from the centroid to its nearest edge point along a predefined angle is then stored. The angles are varied in the intervals of ten degrees and the line is repeatedly extracted. The dimension of each vector equals the number of axes being projected from the centroid. The vector is then normalized to ensure that the vector is scale invariant. Figure 3 shows an example of the normalized shape features extracted from human being while walking. The series of these features encode the action being performed.

Let *S* be the segmented object region within the frame, l_i be a line projected from the centroid to the object boundary at angle *i* to the horizontal line passing through the centroid of the object, then the length of each line to the contour boundary of the object is given by:

$$l_i = \sum_{k=c}^{w} \delta(p(k,l)) \tag{11}$$

k and *l* are the co-ordinates in the *x* and *y* directions respectively, *c* is the centre point and *w* is the contour boundary. *l* is given by $k\tan(\theta)$, $\delta(.)$ is a binary function that returns 0 or 1.

$$\delta(p(k,l)) = \begin{cases} 1 & if\ p(k,l)\ \epsilon\ S \\ 0 & otherwise \end{cases} \tag{12}$$

where p(k, *l*) is the pixel value of the object. The number of lines in each image containing an object as well as the number of neurons in the input layer is j where j = 1, 2, 3, …., n and n is given by n = 360/θmin where θmin is the smallest of the angles. Angular size of 10 degrees interval is used to obtain 36 regions beginning with 10^0. Thirty two (32) of these regions were selected as feature vectors.

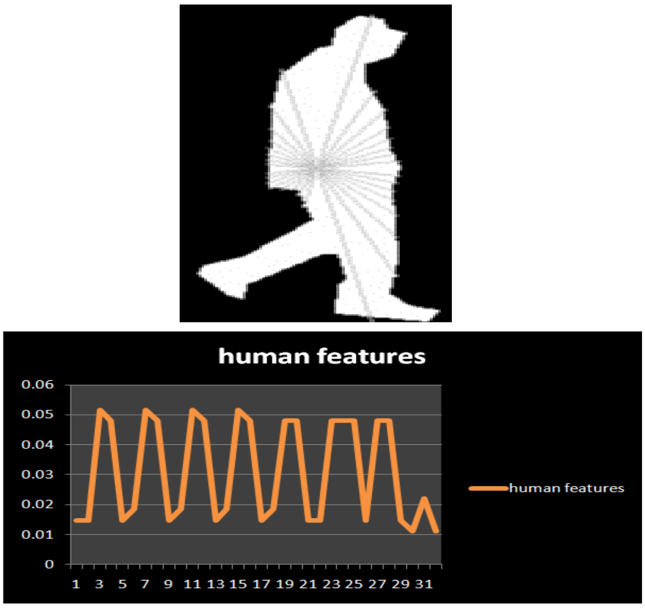

Figure 3 A Typical Silhouette with Associated 1D Distance Signal

Feature Quantization

In this research paper, the discrete Hidden Markov Model is adopted for action recognition. There is, therefore, the need to transform the features to series of codes. This is termed feature quantization. The feature quantization is carried out using K-Means clustering algorithm, which classifies the gesture pattern into k clusters in the feature space (Elmezain *et al.*, 2009). The algorithm is based on the minimum distance between the center of each cluster (centroid) and the feature space. The set of feature vectors are partitioned into a set of clusters. This allows action trajectory to be modeled in the feature space by the clusters. The calculated cluster index is used as input, that is, as the observation symbol to the HMMs. In order to specify the number of cluster *k* for each execution of the Kmeans algorithm, the parameter *k* = 10 is considered, which is based on the cluster number that gives the best performance experimentally.

Figure 4a shows the features extracted from a series of postures representing a complete action. The action in the figure is bending action performed by an actor. Figure 4b shows the features extracted from a series of postures representing a complete action. The action in the figure is jump in place action.

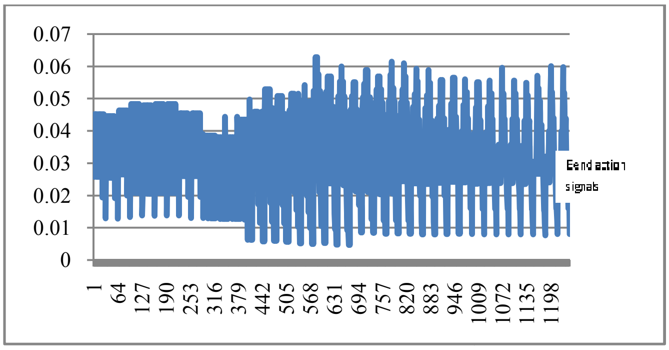

Figure 4a Bend Action Distance Signals

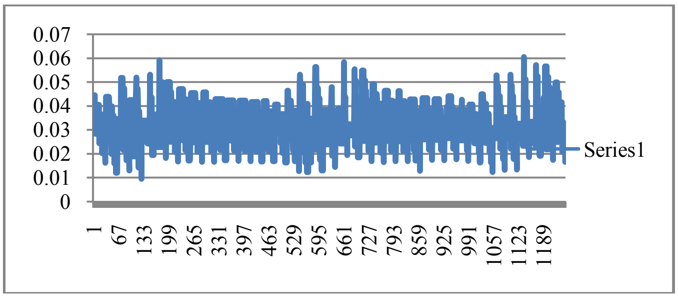

Figure 4b Jump in Place Action Distance Signals

Figure 5a shows the quantization result of bending action after the features have been quantized using Kmeans clustering algorithm. Figure 5b shows the quantization result of jump in place action after the features have been quantized using Kmeans clustering algorithm. It is observed from Figures 5a and 5b that the trajectories of similar actions are very similar with little variances while the trajectories of different actions are dis-similar. Thus actions can be discriminated using these features.

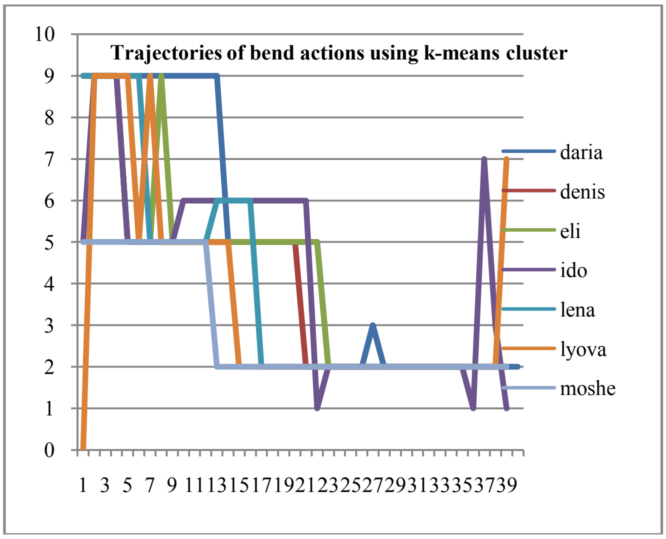

Figure5a Trajectories of Bend Action

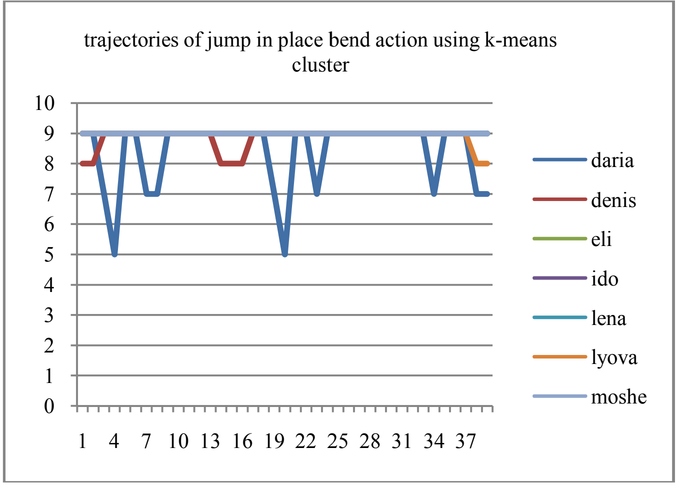

Figure 5b: Trajectories of Jump Actions

Train the HMM Model

HMM is initially used for recognition of voice or signature. However, it is recently used for sequential image recognition such as gesture recognition (Elmezain 2009, gait recognition (Akintola and Tavakkolie, 2012) and activity recognition (Ghazvininejad *et al.*, 2001). A Markov process is a stochastic process where the future event depends on the immediate preceding event. A Markov process assumes that the probability of the occurrence of next event depends only on the occurrence of the current event (Rabiner, 1989).

Hidden Markov Model (HMM) is used to model the action spotting network because it can represent non-action patterns and reflect spatio-temporal variances very well. It has been the most successful and widely used approach to model events which have spatio-temporal variances (Akintola and Tavakkolie 2012; Lee and Kim 1999; Elmezain *et al*. 2009). HMM can be used to estimate the probability of the similarity of an input pattern with a reference pattern. The matching process of the HMM does not require additional consideration for reference patterns with spatial and temporal variances because they are internally represented as probabilities of each state and transition (Yang *et al*. 2007). In addition, the set of unknown patterns can be modeled using a threshold model. In building the threshold model however, it is not easy to train the garbage model that can best match non-action, that is, unknown patterns because the set of non-action patterns is not finite. To overcome this problem, internal segmentation property of the HMM is adopted and a threshold model that consists of states in trained action models and that helps to determine the reliability of the matching results of action models is used as in (Lee and Kim 1999; Elmezain *et al.* 2009).

The internal segmentation property implies that states and transitions in trained HMM represent sub-patterns of an action and a sequential order of sub-patterns implicitly. With this property, a model that can match new patterns generated by combining sub-patterns of action in a different order can be constructed. By constructing a fully connected ergodic model by using states in the action models, a model which can match all patterns generated by combining sub-patterns of an action in any order can be constructed. Figure 6 shows the threshold model that includes two null states, Start State (SS) and Final State (FT). They are null states since they do not emit any observations. A left-right model is constructed and trained for each action model using the Baum-Welch algorithm.

Reduce the number of states in the Threshold Model.

Figure 6 shows the Key action models (up) concatenated with the threshold model (down). The threshold model consists of all the states of key action models which help to determine the reliability of the matching results. In the figure, there are six key action models which are bend, jack, jump in place, run, walk and hand wave. The last model called the threshold model in the figure is obtained by concatenating all the states in the six action models.

A threshold model is constructed by using all the action models parameters. In the new model, each state can reach all other states in a single transition. Observation probabilities of each state and its self-transition in the new model remain the same as in action models and probabilities of outgoing transitions are equally assigned using the fact that the sum of all transition probabilities is 1.0 in a state.

Maintaining the probabilities of states and their self-transitions makes the new model represent all sub-patterns of reference patterns and constructing an ergodic model makes it match well with all patterns generated by combining sub-patterns of reference patterns in any order. Nevertheless, an action can best match an action model because the outgoing transition probability of the threshold model is smaller than that of the action model. Therefore, the output of the threshold model can be used as an adaptive threshold for that of an action model. This is usually called a “threshold model” (Lee and Kim 1999).

After training action models and creating the threshold model, the Action Spotting Network (ASN) for spotting actions from continuous body motions is constructed as shown in Figure 6. The non-gesture model is a weak model for all trained action models. It represents every possible patterns where its likelihood is smaller than the key models for a given meaningful action because of the reduced forward transition probabilities. The likelihood of the threshold model provides a confidence limit for the other actions models.

It is observed that the number of states for threshold models increases as the number of actions increases. There are many states in the threshold model with similar probability distribution which leads to a waste of storage and computational time and complexity. To solve this problem, the symmetric relative entropy between distributions is adopted (Cover and Thomas 1991). The relative entropy measures the distance between two probability distributions. Consider two random probability distributions:

p = $(p_1, p_2, p_3,.., p_N)^{\mathrm{T}}$ and Q = $q_1, q_2, q_3,.., q_N)^{\mathrm{T}}$
The symmetric relative entropy is given by:

$$D(P\|Q) = 1/2\left(\sum_{t=1}^{N}(p_i log\frac{p_i}{q_i} + q_i log\frac{q_i}{p_i})\right) \tag{13}$$

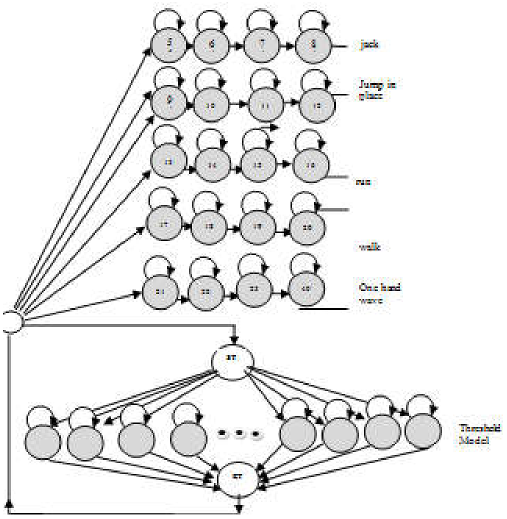

Figure 6 Left-to-Right HMM Model

Figure 7 shows the result of applying the reduction algorithm to reduce a 24 state threshold HMM model to 12 states threshold HMM model. The forward probabilities computed within a window of 16 frames of action sequences by a threshold model are plotted against time. Each of the six actions are modeled using HMM with four states. Therefore, the threshold action model has 24 states. The computation of the probability values of the threshold model takes much time because of the large number of states. The reduction of these states from 24 to 12 is carried out and the probability values of each reduced states is plotted against time as shown in Figure 7. It can be seen that the values obtained from the threshold

HMM of 20 states is very close to that of 24 states, thus 20 states threshold HMM can be used instead of that of 24 states. Although there is a little loss in accuracy but computational time gain is achieved.

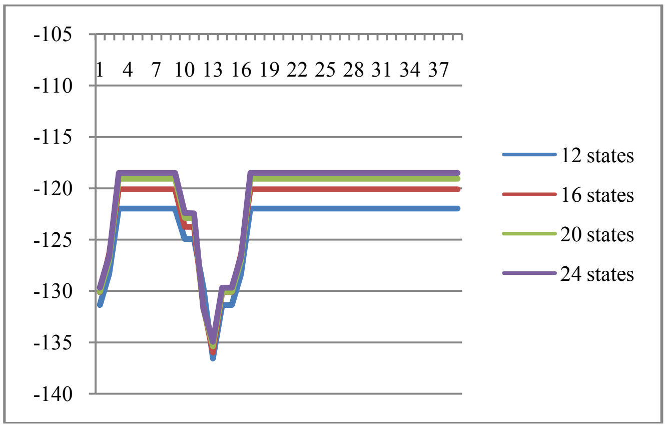

Figure 7 Threshold HMM Values Showing the Approximation from 24-12 States

Action Recognition

The proposed action spotting system contains segmentation module and recognition module. In the action segmentation module, sliding window which calculates the observation probability of all action (key) models and the threshold model are used. The start point and endpoint of an action is spotted using competitive differential observation probability value between the maximal action model (λ_a) and the threshold model (λ_t) (Elmezain *et al.* 2011). When this value changes from negative to positive, it signifies the start of an action and when it changes from positive to negative, it signifies the end of an action.

$$\text{If} \ni a: P(o|\lambda_a) > P(o|\lambda_t) \qquad (15)$$

$$\text{If} \ni a: P(o|\lambda_a) < P(o|\lambda_t) \qquad (16)$$

Figure 8 shows the action spotting network system. It comprises of the feature extraction, the parallel action model network, the maximal action probability module and competitive differential observation probability.

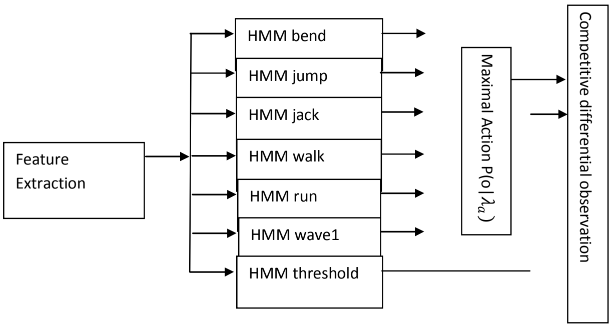

Figure 8 Architecture of Action Spotter

After spotting the start point of an action, in a continuous action sequences, then it activates action recognition module which performs the recognition task for the segmented part accumulatively until it receives the action end signal. To do this, the type of action is recognized using the Viterbi algorithm frame by frame. The Viterbi algorithm of the Viterbi on action recognition of first the quantity $\delta_t(i)$ defined using:

$$\delta_t(i) = \max[\,q_1, q_2, q_3 \ldots . q_t = i,\ O_1\ O_{2,}\ \ldots\ldots O_t|\lambda] \ \ldots . \qquad (17)$$

This is an optimal state sequence that ends in state i, at time t. while accounting for all the appropriate observations. Given this optimality variable $\delta_t(i)$ at time t the optimality variable at time t+1 can be calculated using induction as:

$$\delta_{t+1}(j) = \max[\,\delta_t(i)\ a_{i,j}].\, b_j(O_i) \qquad (18)$$

The optimality variable is initialized using:

$$\delta_1(i) = \pi_i b_i(O_1) \qquad 1 \le\ i \le N \quad . \qquad (19)$$

$$\varphi_1(i) = 0 \ldots . \qquad (20)$$

The recursion is defined as:

$$\delta_t(j) = {}_{1\le i\le N}^{max}[\delta_{t-1}(i)a_{i,j}].\, b_i(O_t) \qquad 2 \le\ t \le T \quad , \quad 1 \le\ j \le N \qquad (21)$$

$$\varphi_t(j) = {}_{1\le i\le N}^{argmax}[\delta_{t-1}(i)a_{i,j}] \qquad 2 \le\ t \le T \qquad 1 \le j \le N \qquad (22)$$

where $\varphi_1(i)$ keeps track of the states that maximized Equation 22. The termination criterion is defined using Equation (23) and Equation (24)

$$\text{P} = {}_{1\le i\le N}^{max}[\delta_t(i)] \qquad (23)$$

$$\text{S}_\text{T} = {}_{1\le i\le N}^{argmax}\ [\delta_t(i)] \qquad (24)$$

The optimal state sequence is obtained by backtracking through the $\varphi_1(i)$ matrix.

$$\text{S}_\text{t} = \varphi_{t+1}(S_{t+1}),\ \text{where t} = \text{T} - 1,\ \text{T} - 2, \text{T} - 3\,, \ldots . , 1 \qquad (25)$$

Experiments on Activity Recognition

The model is tested on the Weizmann dataset. The dataset contains ninety videos of ten main actions performed by nine different people. Only six actions performed by eight different people were used for experiment. Example frames of this well known dataset are shown in Figure 9. Videos of four subjects were used to train each of the six HMM models using Baulm-Welch algorithm. Feature vectors are extracted as 1-D distance signals of the silhouettes from these videos as discussed in Section 3.3. To quantize these signals, K-Means clustering algorithm is used. Each action is modeled using a four state Left-to-Right Hidden Markov Model. The initial probabilities of each of the HMM are set to {0.25, 0.25, 0.25, 0.25} while the state transition probabilities and the observation probabilities are randomly selected. To train the HMM, Baum-Welch algorithm is adopted. A forward algorithm is then applied to the action sequence to obtain the probability of the model given the sequence. This is repeated on all the models and the model that maximizes the probability is recognized as the action that the sequence represents.

Figure 10 shows the convergence of Baum-Welch algorithm on six action sequences. As the models parameters become optimal, the probability change tends to remain constant from iteration to iteration. Six different action features vectors are stacked together to test the action segmentation algorithm. A sliding window of sixteen frames is used to calculate the forward probability of the action. This is because it takes an average of sixteen frames or more to complete an action. The maximal action probability is calculated by finding the maximum forward probabilities of each action in each sliding window. The forward probability of the sliding window of the threshold action HMM model is also calculated. A competitive differential probability between the maximal action HMM and the threshold HMM is then obtained. The start of an action begins where the maximal action HMM value is greater than the threshold HMM value and ends where the maximal is less than the threshold value or any other action HMM value.

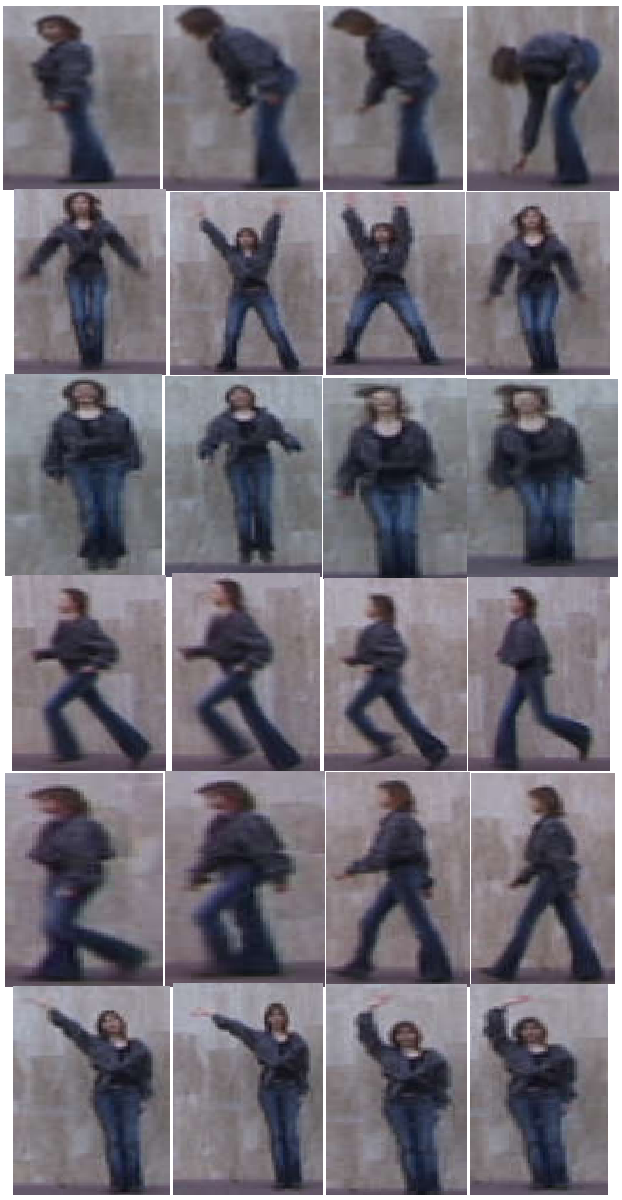

Figure 9 Sample Actions from the Weizmann Dataset for Training

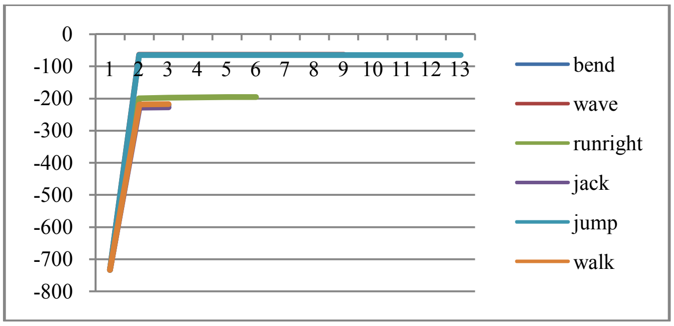

Figure 10 Model1-Model6 Baulm-Welch Convergence Action Training Graphs

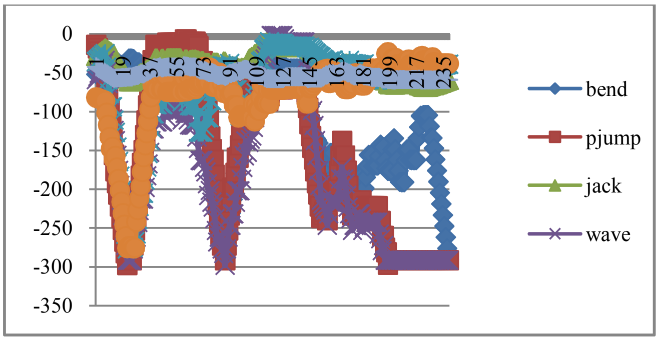

Figure 11a Results of Window-Based P(O|M) for Person 1

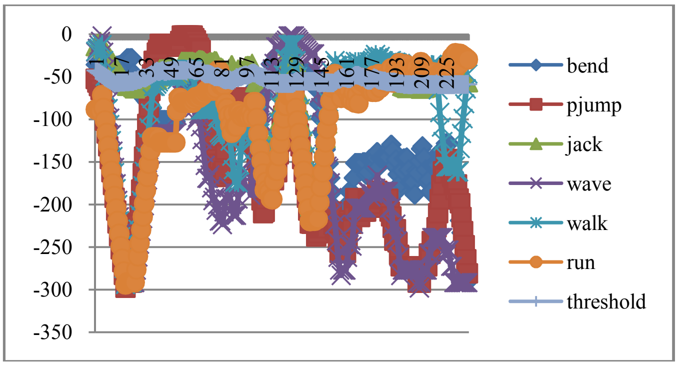

Figure 11b Results of Window-Based P(O|M) for Person 2

Figure 11a and Figure 11b show the result of the forward HMM algorithm on concatenation of different actions performed by a particular person. As the detection of a key action pattern begins, the probability of that action gets bigger over all other actions. The threshold will now detect whether that sequence is an action if its probability is greater than that of the threshold, thus the threshold is a measure of adaptability to the action recognition scheme.

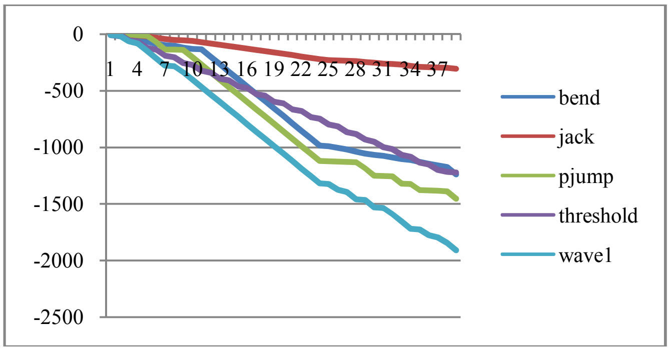

Figure 12 Viterbi Decoding of Jack Action Sequence

Figure 12 shows the Viterbi decoding of five action segments that were passed from the segmentation algorithm. It is observed that the Viterbi probability of Jack action is the highest, thus the segment is thus classified as Jack action.

Figure 13 shows the result of action segmentation using the proposed threshold HMM model on the experimental data. It is shown that bend, run, walk and wave1 are 100% recognized while Jack is 84% recognized and bend is 75% recognized. The recognition scheme is highly sensitive to the quality of the silhouette extracted from the background.

The execution times of different modules on dual core system are computed in order to know the processing time threshold. Figure 14 shows the processing times of the selected modules. It is observed that action recognition takes the largest execution time (1.4ms) followed by the background subtraction (1.1ms) followed by object tracking (0.8ms). Since these modules can be made to operate in parallel fashion on multi-core machines using parallel programming concept, it can be concluded that the system threshold is action recognition time which is 1.4Ms. The processing times of these modules however show that the system can be applied in real-time surveillance scenarios which normally require very low processing time.

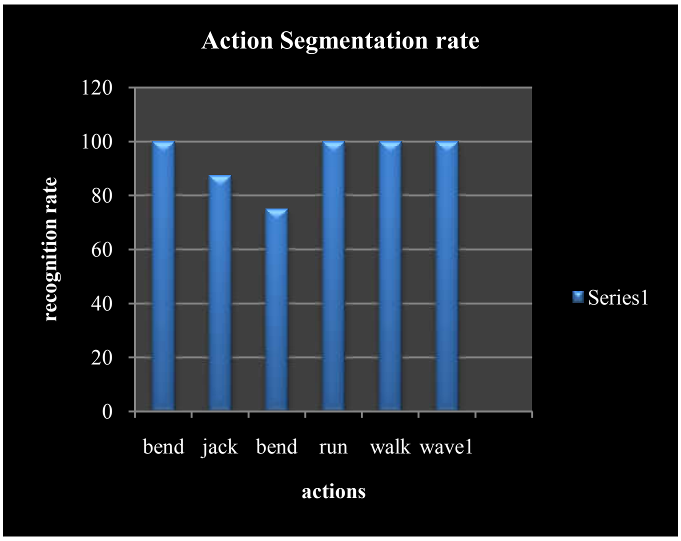

Figure 13 The Results of Action Segmentation for all Persons.

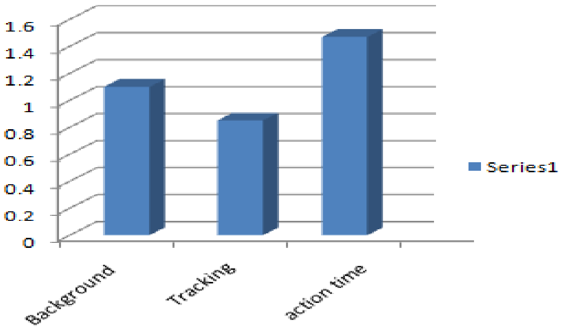

Figure 15 Processing Times of the Modules

CONCLUSIONS

The subject of identifying humans and their activities in real-time videos have been the research focus in computer vision community for the past few decades. An integral part of video activities recognition is the ability to segment the human object from the dynamic environments. A lot of work has been done in the area of segmenting foregrounds from background scene. These methods include Optical Flow, Mixture of Gaussians (MOG) and Eigen-Backgrounds. Most of these schemes, apart from being computationally expensive are not adaptive to quasi-stationary backgrounds. In this paper, an adaptive threshold based kernel density estimation combined with spatio-temporal tracking algorithm is adopted to solve the problem of foreground segmentation and tracking in quasi-stationary backgrounds. A good representation of the foreground object is a key to recognition accuracy and timely activity recognition. This paper have reported on the use of radial signals distance features which is computationally efficient. Automatic action segmentation in real time video scenes is also a challenge. This is because actions vary in spatio-temporal duration and in viewing angles. The activity recognition part of this work constitutes a significant contribution to the field of automated behavior recognition. Human behaviours, as formulated in this paper have not been addressed in the related literature. Hidden Markov Models is used for detecting some predefined activities by using Radial signal distance signals which is very fast to calculate in real time and therefore useful in real time action recognition in videos. This research has several challenges such as adaptability to changing environment, view invariant activity segmentation and lack of efficient foreground detection. These challenges will be addressed in future.

References

1. Akintola K.G, Akinyokun O.C, Angaye C.O and Olabode O. (2016). Vehicles and Pedestrians Classification using Radial Shape Descriptors for Video Surveillance. *International Journal of Artificial Intelligence Research. Vol. 5, No. 2, Pages xxx-xxx, Published by Artificial Intelligence Research, Canada.*
2. Akintola K.G, Tavakkolie A. (2011). Robust Foreground Detection in Videos using Adaptive Colour Histogram Thresholding and Shadow Removal. *Proceedings of the 7th International Symposium on Visual Computing*, *Las-Vegas, NV, Pages 496-505.*
3. Akintola K.G, Tavakkolie A. (2012). A Novel Gait Recognition System Based on Hidden Markov Models. *Proceedings of the 8th International Symposium on Visual Computing Crete Greece, Pages125-134.*
4. Akintola K.G. (2014) Real-Time Object Detection and Tracking for Video Surveillance. *International Journal of research in Computer Applications and Robotics, Vol. 2, No.7, Pages 161-173.*
5. Bashir F.I, Khokhar A.A, Schonfeld D. (2007). Object Trajectory Activity Classification and Recognition using Hidden Markov Models. *IEEE Transactions on Image Processing, Vol. 16, No. 17, Pages 1912-1919.*
6. Bobick A.F. and Davis J.W. (2001). The Recognition of Human Movement using Temporal Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence *(T-PAMI), Vol. 23, No 3, Pages 257- 267.*
7. Chen, D., Wactlar, H, Chen, M., Gao, C., Bharucha, A., Hauptmann, A. (2008). Recognition of Aggressive Human Behaviour using Binary Local Motion Descriptors. Proceedings of *30th Annual International IEEE Engineering in Medicine and Biology (EMBC) Conference, Vancour, British Columbia, Canada, Pages 5238-5241.*
8. Chen M., Hauptmann, A. (2009). MoSIFT: Recognizing Human Actions in Surveillance Videos. *Technical Report. Carnegie Mellon University, Pittsburgh, USA.*
9. Comaniciu D., Ramesh V., and Meer P. (2000). Real-Time Tracking of Non-Rigid Objects using Mean Shift. *Proceedings of Conference on Computer Vision and Pattern Recognition*, Pages *II:142–149, Hilton Head, SC.*
10. Cover, T.M., Thomas, J.A. (1991): Entropy, Relative Entropy and Mutual Information. *Elements of Information Theory, Pages 12-49.*
11. Dollars P., Rabaud V., Cottrell G., and Belongie S. (2005). Behavior Recognition via Sparse Spatiotemporal

Features. *Proceedings of VS-PETS Conference. Beijing China, Pages 65-72.*

12. Elgammal A., Duraiswami R, Harwood D and Davis L.S. (2002). Background and Foreground
13. Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance. *Proceedings of the Conference of IEEE Vol. 90, No.7, Pages 1151-1163.*
14. Elmezain, M; Al-Hamadi, A and Michaelis, B. (2009). A Novel System for Automatic Hand Gesture Spotting and Recognition in Stereo Color Image Sequences. *Journal of WSCG'09, Vol. 17, No 1, Pages 89-96, ISSN 1213-6972.*
15. Ghazvininejad M., Rabiee H.R., Pourdamghani N., Khanipour P. (2011). HMM Based Semi-Supervised Learning for Activity Recognition. *Proceedings of the 2011 International Workshop on Situation Activity and Goal Awareness, Pages* 95-100.
16. Gonzalez J, Varona X , Villanueva J, Roca F. (2001). Online Human Activity Recognition for
17. Video Surveillance. Proceedings of ix National Symposium on Pattern Recognition and Image Analysis (SNRFAI), *Castellon de la Plana, Spain.*
18. Jain A. K., Ross A. and Prabhakar S. (2004). An Introduction to Biometric Recognition. *IEEE*
19. *Transactions on Circuits and Systems for Video Technologies, Vol. 14, No. 1, Pages 4-20.*
20. Klaser (2010). Learning Human Actions in Videos. *PhD Thesis, Department of Mathematics and Informatics, University of Grendole.*
21. Laptev I. (2005). On Space-Time Interest Points. *International Journal of Computer Vision Vol. 64 No. 2, Pages 107-123.*
22. Lee H.K. and Kim J.H. (1999). An HMM-Based Threshold Model Approach for Gesture recognition. *IEEE PAMI, Vol. 21, No.10, Pages 961–973.*
23. Malgireddy M.R. Corso J.J., Setlur S. (2010). A Framework for Hand Gesture Spotting using Sub-gesture Modeling. *Proceedings of 20th IEEE International Conference on Pattern Recognition (ICPR 2010), August 23-26, pp. 3780-3783, Istanbul, Turkey.*
24. Perez P., Hue C., Vermaak J., and Gangnet M., (2002). Color-Based Probabilistic Tracking. *Proceedings of European Conf. on Computer Vision, (ECCV), May 28-31, Vol. 1. Pages 661-675, Copenhagen, Denmark.*
25. Rabiner L.R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE, Vol. 77 No. 2, Pages 257-285.*
26. Yang H., Park A., Lee S. (2007). Gesture Spotting and Recognition for Human–Robot Interaction. *IEEE Transactions on Robotics*, Vol. 23 No 2, *Pages 256-270.*

How to cite this article:

Akinyokun O. C and Akintola K. G (2018) 'Real-Time Whole-Body Action Recognition in Videos Using Threshold Hidden Markov Model ', *International Journal of Current Advanced Research,* 07(5), pp. 12248-12257.
DOI: http://dx.doi.org/10.24327/ijcar.2018.12257.2146

****