Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# A GESTURE RECOGNITION SYSTEM FOR GESTURE CONTROL ON INTERNET OF THINGS SERVICES

[1]TALAL H. NOOR

[1] Assistant Professor, College of Computer Science and Engineering, Taibah University, Yanbu, Medinah 46421-7143, Saudi Arabia.

E-mail: [1]tnoor@taibahu.edu.sa

**ABSTRACT**

Internet of Things (IoT) is a promising computing model, which uses several enabling technologies to provide new type of smart services that allow users to interact with daily objects in a different way. Most of IoT services are invoked using touch screens and connected to Smartphones or tablets to enable a real-time connectivity between users and things. However, only a few IoT systems consider gesture based user interactions for their services which allow users to have a better experience with IoT products. In this work, we present a gesture recognition system for gesture control on IoT services. In particular, the gesture recognition system is based on alphabet characters and number to classify the hand fingertip trajectory using a hidden Markov model. The presented system composed of three key stages. Firstly, color information with 3D depth information that segment the correct position of hands. Then the fingertip is detected based on the curvature of hand contour. Secondly, the dynamic features in polar coordinates are employed using k-mean clustering technique. Finally, Baum-welch procedure is used to carry out the learning gestures and the viterbi algorithm is used to recognize the gesture. The experiments demonstrate that the presented system has a capability to classify the isolated gesture dues to spatio-temporal variability. Precisely, our experiment provides a promising result where the average recognition rates achieved 98.61% and 93.06% for training and testing dataset, respectively. Furthermore, it provides finer presentation and low-slung computational difficulty when functional image sequences samples of multipart circumstances are used.

**Keywords:** *Internet of things' services, gesture recognition, gesture control, color information, depth map, hidden Markov model.*

## 1. INTRODUCTION

In the last decade, Internet of Things (IoT) has attracted researchers and practitioners to work on this new computing paradigm [1], [2]. IoT uses several enabling technologies like sensors, Radio-Frequency IDentification (RFID), mobile computing and cloud computing to allow interactions between IoT products (i.e., things or devices) and users [3]. The interactions include notifications from the IoT products which give users information about the IoT products (e.g., an air conditioner send the user a notification that it has been left open while nobody is there) which sometime require actions (e.g., the user can use shutdown action to shutdown the air conditioner) from the user.

Most of IoT products provide services where interactions with users are handled using touch screens and connected to Smartphones or tablets to enable a real-time connectivity between users and IoT products. However, only a few IoT systems consider gesture based user interactions for their services which allow users to have a better experience with IoT products [1]. Because gesture recognition systems are complex and has several issues that should be tackled, notably for real-time application of Human-computer Interaction (HCI). A spatio-temporal problem forms challenge in recognizing hand gesture (i.e., A gesture is constituted from the hand motion trajectory while posture is static morphs of hand), since the gesture can differ in silhouette, duration and path even for an individual. Other problems include what and how gesture features can be extracted, either for learning or testing processes. According to the hand motion enclosure, the motion may be classified into two classes; gesture and posture as shown in Figure 1.

The first one is named gesture, which built by the motion of fingertip to construct the meaning. But, the position of the static hand, which keep on the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

same space is called posture. The major motivation in the wake of using fingertip gesture is to formulate the interaction between the user and the IoT product (i.e., device or computer) which is practically the same when dealing between a user and another user. This will allow users to efficiently exploit IoT features which represent the main objective of this study. So, one of the famous applications in HCI is sign language recognition which allows the interaction between a person and a computer. To classify sign language, there are three main groups; word-level symbol, finger spelling and non-manual structures [4]. It is being the major interaction within non-manual features and word-level sign that includes a mimic expression like mouth and body position. To understand the vital issues of research most used techniques of gesture and posture application are accurately addressed. So, finger spelling is used to deal with sentence letter by letter. The adjustment will assist us in the recognition process, which makes the research more truthful in keeping with the rate of recognition. Moreover, good segmentation and good features extraction for the Region Of Interest (ROI) represent a fundamental part of the recognition task.

*Figure 1: The first two samples represent the posture and the other samples refer to the gesture.*

So, we present a system for gesture recognition to control IoT services and enable a real-time connectivity between users and IoT products which allow users to have a better experience with IoT products. In particular, the gesture recognition system probe motion trajectory so-call gesture path, which is generated using fingertip detection to transaction an application for recognizing numbers 0 to 9 and alphabets A to Z. The proposed system has the capability to deal with complex scenes by using color information and 3D depth information to alleviate the problems of brightness variation and ROI overlapping. The dynamic features in polar coordinates are employed using k-mean clustering technique. Therefore, Baum-welch technique is used to carry out learning process and the viterbi algorithm is used to recognize the gesture. The experiments demonstrate that the presented system has a capability to classify the isolated gesture dues to spatio-temporal variability. It is noted that, the presented system achieves superior recognition in a multifaceted scene which hold various situation as overlap and incomplete occlusion between hand and face regions.

The remainder of the article is structured as the following. The previous research is introduced in Section 2. Section 3, presents the architecture system of gesture recognition for control on IoT services. The segmentation and tracking of the AOI is presented in Section 4. Section 5, formulates the polar features of motion trajectory. In Section 6, we describe the Hidden Markov Model (HMM) technique. Section 7, carries out the experimental results. At the end, we conclude our work in Section 8 and discuss some future work.

## 2. RELATED WORK

The significance of IoT has been recognized by many research works [1]-[3], [5]. Want et al. [6] highlighted that IoT provides users with the capability to control and monitor things (i.e., devices) using Internet technologies. Some research works focused on gesture control on IoT. For example, Han and Rashid [7], proposed a system of combined voice and gesture control of internet of things. In particular, the system consists of two stations namely control and appliance. Both stations are developed using Zigbee module. For the gesture recognition, the authors used border following, convex hull, and ramer-douglas-peucker algorithms.

One of the crucial applications in gesture recognition is the sign language recognition which makes the interaction between the user and IoT product (e.g., device or computer) achievable. There are various techniques used in the literature for classifying and recognizing sign languages like alphabets and numbers. For instance, one of the techniques used is the Adaptive Neuro-Fuzzy Inference System (ANFIS) which was developed for recognizing Arabic Sign Language (ASL) [8]. Here, the segmentation problem is alleviated using colored gloves which facilitate the system to take out the finest features. But, Handouyahia et al. [9] introduce an approach to classify and recognize International Sign Language (ISL) via Neural Network (NN) to train and test alphabet characters. The most important advantage of using NN is to without doubt test and learn the extracted features commencing the region of interest (i.e., sign language). In 3D hand posture, the feature of Elliptic Fourier Descriptor (EFD) provides a foremost task for recognition [10]. In addition, this method measures the angles and silhouettes for the hand region as key features for classification. Licsar and Sziranyi [11] engaged the coefficients of

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Fourier to precede the shape of hand region to recognize the gestures. Furthermore, Freeman and Roth [12] recognize alphabets character using the histogram of hand orientation. This method was to ease the misclassification problem by using a huge dataset for learning to indulgence the orientation problems.

Unlike previous work, we propose a gesture recognition system for gesture control on IoT services. In particular, the gesture recognition system is based on alphabet characters and number to classify the hand fingertip trasjectory using a hidden Markov model. Moreover, unlike previous work such as [7], our approach is able to recognize hand gesture even if the background has colors similar to the face and hands. This is because of Gaussian Mixture Model (GMM) which is learnt using skin and non-skin datasets. Furthermore, our experiment provides a promising result where the average recognition rates achieved 98.61% and 93.06% for training and testing dataset, respectively. Furthermore, it provides finer presentation and low-slung computational difficulty when functional image sequences samples of multipart circumstances are used.

## 3. SYSTEM ARCHITECTURE

In this section, we overview the design and architecture of our gesture recognition system for gesture control on IoT services to allow users to have a better experience with IoT products. In particular, our architecture is based on the idea of combining both IoT with cloud computing to benefit of the CloudIoT paradigm [5]. Figure 2 shows the key components of the system architecture which consists of four different layers namely: i) Things (devices) layer, ii) Mobile network layer, iii) Gesture recognition layer, and iv) Cloud services provider layer.

*Things (devices) layer.* This layer consists of many IoT products (i.e., things or devices) that are supported with sensors, vision and communication technologies to allow users to monitor, control and access the IoT products' information. These IoT products use communication technologies to connect to the mobile network layer.

*Mobile network layer.* This layer consists of several mobile network technologies such as Wireless Access Points (WAPs), Base Transceiver Station (BTS), or satellite. These technologies allow IoT products to connect to the Internet for sending notifications or receiving action requests from users.

*Gesture recognition layer.* This layer consists of the IoT user where hand movements are tracked by the camera. Moreover, the layer consists four components namely fingertip detector, feature extractor, gesture recognizer, and service invoker. The fingertip detector component is responsible for detecting the fingertip based on the curvature of hand contour. The feature extractor component is responsible for extracting the dynamic features in polar coordinates which are employed using k-mean clustering technique. The gesture recognizer component is responsible for recognizing the hand gesture using the viterbi algorithm. Last but not least, the service invoker component is responsible for triggering the IoT service based on the recognized hand gesture. More detailed about the aforementioned components will be explained in the following sections.

*Cloud services provider layer.* The layer contains many cloud service providers which provide cloud services like Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS). Based on the IoT product requirements such as storage or computation, the cloud service will be provisioned on demand. For instance, the gesture database can be stored in the cloud using IaaS.

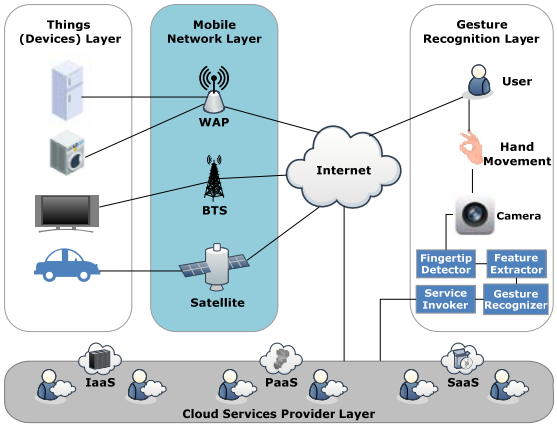



*Figure 2: Architecture of the Gesture Recognition System for Gesture Control on IoT Services.*

## 4. SEGMENTATION AND FINGERTIP DETECTION

To correctly segment the hands from stereo color images with complex scene, color information and 3D depth takes place using $YC_bC_r$ color space in conjunction with Gaussian Mixture Model (GMM) and Bumblebee stereo camera. At first, the

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

segmentation of skin colored areas be forceful when we use the chrominance only in analysis. So, we use $YC_bC_r$ in which $Y$ channel is to illumination and ($C_b$,$C_r$) represent chrominance. To diminish the effect of brightness variation, $Y$ channel is disregarded and the chrominance channels are only used because of full color info. The GMM is learned using a skin and non-skin dataset, which is accumulated from the internet with 35 different race subjects for skin pixels and 85 unlike images for non-skin pixels. The GMM technique starts with skin model using skin dataset in which an abnormal k-means clustering procedure [13] makes learning process to identify the original GMM parameters. In addition, the algorithm takes place to segment skin regions for face and two hands in video color. It computes the depth value $Z$ as well as the skin color info (Eq. 1 and (Figure. 3(a)) [14].

$$Z = \frac{f.b}{x_L - x_R} \quad (1)$$

Such that $f$ refers to the alike effective focal length and b represents the base line, which measure the distance between two optic centers; left and right ($O_L$, $O_R$). Additionally, the orientation subtended via two optic axes is to $2\theta$. A point $P(X, Y, Z)$ in 3D space is anticipated according to two points ($x_R$, $y_R$) and ($x_L$, $y_L$) for image plane of right and left camera. By using the cross correlation and calibration data of cameras, the depth value can be obtained by passive stereo measuring for the results 3D-points from many clusters. The clustering technique is to measure the area rising with two criteria's in 3D; skin color and Euclidean distance. It is being noted that this method is supplementary powerful to incomplete occlusion and disadvantageous lighting, which stand up to real-time requests (i.e., cloud gesture recognition for case in point).

Using the depth value for the camera set-up system (Figure. 3(d)), the overlap problematic between face and hands is frozen since the hand area are close to the camera position rather than the face area.

The hand's contour acts a significant role in the detection of fingertips. For every pixel of hand's contour, the neighbor points in this contour are used to estimate the k-curvature [15]. The curvature is used to calculate the pattern boundary point at $k$. The main idea is that the fingertips are marked when the contour point take place with values above the ground curvature (i.e., potential peaks). Here the curvature is to the ration between length $l$ and displacement $d$. $L$ is the outline of all distances contained in shape curve. A displacement $d$ represents the distance between the first and latter contour points. The curvature is estimated by the following equation:

$$k-curvature = \frac{1}{d} = \frac{\sum_{i=k-n/2}^{k+n/2} \|p_i - p_{i+1}\|}{\|p_{k-n/2} - p_{k+n/2}\|} \quad (2)$$

where $n$ represents the whole number of curvature pixels estimation, $P_i$ and $P_{i+1}$ are successive points of the margin of objects.

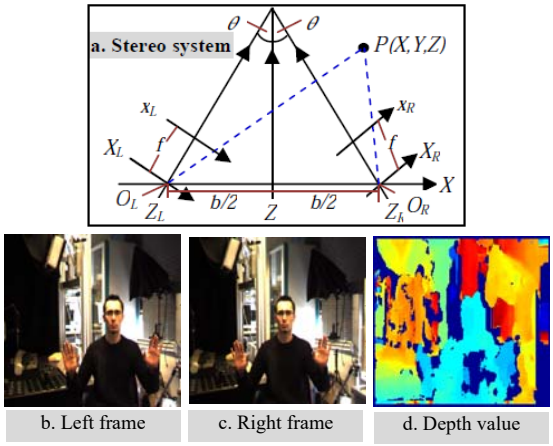



*Figure 3: (a) Stereo System of Camera Geometry. (b) Left Frame of Sequence Image Stream. (c) Right Frame. (d) The Depth Value for Left and Right Fame Using Bumblebee Camera.*

The depth information is estimated for the ROI automatically. For precision, the range of depth value is in-between 40 cm to 180 cm. Furthermore, the peaks in hand's contour, those curvature values on top of minimum threshold, represents the fingertips. The threshold value is Empirically putted between 1 and 4 in our proposed work. Dropping this value increases the false positive rate for the peaks detection. However, ever-increasing this threshold value will allocate an outsized number of peaks to be detected.

As shown in Figure. 4, the two clusters $C_1$ and $C_2$ (i.e., the maximum value for each cluster) are carefully chosen using the local extreme maximum value. It is being noted that the maximum two points are considered as fingertips ($SCP_1$ and $SCP_2$). Even so, the fingertip is detected mistakenly because the proposed technique can consider both peak and valley points as fingertips. To alleviate the problem, the distance between the selected points of contour (i.e., $SCP_1$ and $SCP_2$) and the center

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

point of target object *CP* is calculated as illustrated in Figure 4. In addition, a normalization procedure is taken to scale these points with the range 0:1. As a result, the values of points, which are above 0.5 are classified as fingertips. In the bottom part of the graph in Figure 4, the red point represents the fingertip (peak), whereas the green point refers to a valley.

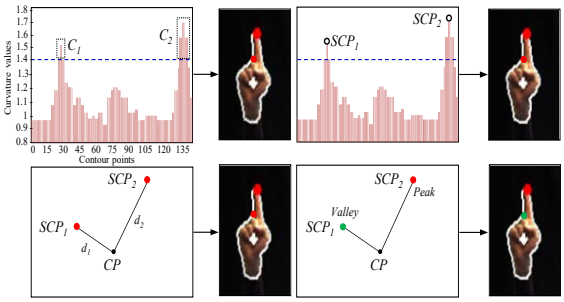
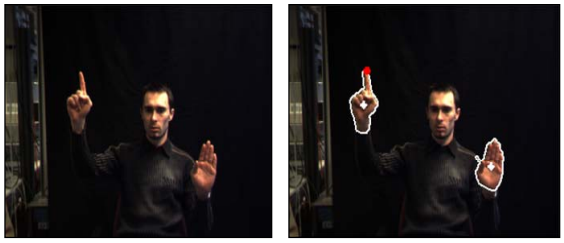



*Figure 4: Detection of Valley and Peak Points. In Above Figure, The Max Local Extreme Value is to Obtain Contour Points $SCP_1$ and $SCP_2$ Using Two Clusters $C_1$ and $C_2$. The Red Color is a Fingertip Detection in Which the Normalization Value is Above 0.5.*

In case of using static background, the proposed technique is well-thought-out as best in stretch of results to detect the fingertips points (Figure 5(b)). Furthermore, the technique considers the scaling difficulties and cures the false classification among neighboring pixels. Additionally, the technique works robustly under occlusion, because it is based on depth information, as well as it is with no cost when compared with previous techniques, which uses histogram to obtain fingertip [16].



(a) Source frame. (b) Detected fingertip.

*Figure 5. The Detected Fingertip is Marked with Red Point. The Centroid Point and Hand's Contour is Marked by White Point.*

In the segmentation stage, the color histogram is computed using Epanechnikov kernel when hand's target is detected [17]. The kernel allocates less weight to pixels far from the center to enhance the density estimation strength. In success consecutive frames, the best match of hand's target is obtained using the help of Bhattacharyya coefficient [18], which computes the similarity using maximizing Bayes error. This error stands up from the relationship of hand candidate and target. Take in consideration the value of mean depth that is computed using preceding hand area to alleviate the intersection between face and hands. Here, the mean-shift procedure is to identify and run the optimization recursively to estimate the vector of mean shifts. This optimization gives measured location for hand's target, in addition to the computed hesitation estimation. These processes are followed by Kalman iteration. Kalman identifies predicated hand's target position. Thus, the gesture trajectory for the hand is calculated using the communications of detected hand in-between consecutive frames. The readers can find more information in the following references [19], [20], [21].

## 4. FEATURE EXTRACTION

The motion trajectory so-called a gesture path is spatio-temporal pattern of hand position (x, y). Here, in Cartesian space the coordinate is extracted using gesture frames and then changed directly to Polar coordinates (see Figure. 6).


(a) (b) (c)

*Figure 6: Gesture Path `R' Transformation from Cartesian Space to Polar Spaces. (a) x-y Coordinates Space for Gesture `R'. (b) $\rho_c\varphi_c$ Polar Coordinates Space of Gesture `R'. (c) $\rho_{sc}\varphi_{sc}$ Space of `R'*

One feature is called a location feature *Lc,* which represents a distance between the center point of hand gesture to all contour gesture points. Another location feature *Lsc* is determined using the start point of hand gesture and the current point of this gesture. In addition, there are three different orientation features, which are based on displacement vector at every point. $\theta_{1t}$ refers to the center of hand gesture path, $\theta_{2t}$ represents the orientation between two sequential points. The orientation $\theta_{3t}$ refers to the angle between the start and the current point of hand gesture. Also, the velocity feature plays a significant role in recognition and is treated as an Euclidean distance between two points over time *t*.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Polar coordinate is computed from Cartesian coordinates directly based on the spatio-tempral points of the path of hand gesture. To determine the gesture path in polar coordinates, the radius from gesture centroid point (Eq. 3) and the radius in-between the start and the current points of hand gesture path are used (Eq. 4.).

$$rsc_{max} = max(Lsc_t),\ \rho_{sct} = \frac{Lsc_t}{rsc_{max}},\ \varphi_{sct} = \frac{\theta_{3t}}{2\pi} \quad (3)$$

$$rc_{max} = max(Lc_t),\quad \rho_{ct} = \frac{Lc_t}{rc_{max}},\quad \varphi_{ct} = \frac{\theta_{1t}}{2\pi} \quad (4)$$

where $rsc_{max}$ is the longest distance vector from start point to each point in hand gesture path. $rc_{max}$ is to the longest distance from gesture centroid point to every point in that gesture path.

In polar coordinate, various combinations of features are extracted to determine a diversity vectors of features (Eq. 5 & Eq. 6). For instance, the feature vector is single-minded at frame $t$+1 using the union of locations features with respect to centroid point for velocity feature ($\rho_{ct}$, $\varphi_{ct}$, $V_t$). Additionally, the locations feature with respect to the start and the current points of hand gesture with velocity feature ($\rho_{sct}$, $\varphi_{sct}$, $V_t$). It is being noted that the final feature vector is obtained by combining all the vectors ($\rho_{ct}$, $\varphi_{ct}$, $\rho_{sct}$, $\varphi_{sct}$, $V_t$). Figure 6 illustrate the depiction of the hand gesture `*R*' with respect to *x-y*, $\rho_c$-$\varphi_c$ and $\rho_{sc}$-$\varphi_{sc}$ coordinates spaces, respectively. Moreover, we note that an observable variance is found in the illustration of gesture `*R*' particularly in $\rho_c$-$\varphi_c$ and $\rho_{sc}$-$\varphi_{sc}$. This variation is vital to determine powerful features for the suggested gesture system.

$$F_c = \{(\rho_{c1}, \varphi_{c1}), (\rho_{c2}, \varphi_{c2}), \ldots, (\rho_{cT-1}, \varphi_{cT-1})\} \quad (5)$$

$$F_{sc} = \{(\rho_{sc1}, \varphi_{sc1}), (\rho_{sc2}, \varphi_{sc2}), \ldots, (\rho_{scT-1}, \varphi_{scT-1})\} \quad (6)$$

The obtained feature vector is quantized to constitute the discrete vector symbol that are employed as an input to the Hidden Markov Model (HMM). Furthermore, we use k-mean clustering algorithm over polar coordinate features to categorize hand gesture feature to *K* cluster on that space. Here, k-mean technique is based on minimum distance between the center for every cluster and hand feature point [22, 23] that models the hand gesture path using various clusters in feature space. Hence, the computed index of cluster is used as an emission values to HMM. In our dataset, the best number of clusters is unidentified frequently. So, the cluster number impartially relies on the segmented numbers theoretically (i.e., every gesture character from A to Z and Number from 0 to 9). In each cluster the straight-line segment is classified to a single cluster. The key advantages of using k-means is to converges faster, representation ease and further scalable.

## 5. HIDDEN MARKOV MODEL CLASSIFICATION

In the classification process, the symbols are assigned to its own classes. Two mains tasks play a very important role to perform learning and testing the symbol. Baum-welch function is employed to train the initialized parameters of HMM $\lambda = (A, B, \pi)$ [24]. Consequently, the classification process matches tested hand gesture in opposition to gestures database (i.e., stored in the cloud) to classify which class will belong to it. In that way, the hand gesture is classified equivalent to maximum likelihood for hand gestures models based on Viterbi function [24]. The maximum gesture means that the gesture model has the highest value between all building gestures models (Figure 7).

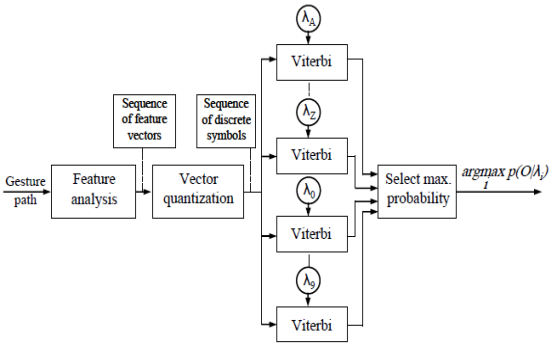



*Figure 7: Block Diagram for Isolated Gestures Using Viterbi Recognizer.*

The size of HMM must be resolute before learning each hand gesture model. Thus, how many states do we require? The number of states relies on the difficulty of every hand gesture in which we divide it to segment parts. Each segment is assigned to one state. Shortly speaking, the number of segmented parts in each hand graphical pattern is considered during the representation. To characterize various hand gesture patterns, we take in consideration the included possible patterns in a gesture and so calculate how many segment parts. For instance, to signify a hand gesture pattern `L',

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

we only need two states, whereas a graphical pattern `E' requires six states as well as four states for hand gesture pattern `3'. It is observed, that the over-fitting issue will occur when there is no sufficient number of learning because of excessive number of states. In other words, over-fitting happens when the HMM provide random error rather than fundamental connection. Powerful over-fitting issue does not base on the number of parameters and data only, but also on the structure of building model compatibility related to the quantity of models and data shapes. To alleviate this issue, additional techniques such as early stopping, regularization, and cross-validation are used when the added learning is not mostly resulting (for further information, the reader can refer to [25]). Additionally, when we use inadequate states number, HMM bias power is reduced because of an extra segmented part to gesture pattern is allocated and demonstrated on one state.

Before preparatory the iterative Baum-welch for learning process, the initial parameters values of HMM must be assigned with respect to Left-right Banded Model (i.e., LRB topology) [24]. The main motivation behind using LRB is to increase the state index or at least to maintain the same state index while time increases. Thus, the structure of LRB cannot be easily lost [26]. Spontaneously, an observation is that, a best HMM parameters initialization ($A$, $B$, $\pi$) realizes worse results. Formally speaking, Matrix $A$ is to transition parameter and is represented via left banded model as follows:

$$A = \begin{pmatrix} a_{11} & 1-a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & 1-a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \quad (7)$$

The diagonal elements $a_{ii}$ of the transition matrix can be selected to provide just about average state durations $d$; such that:

$$a_{ii} = 1 - \frac{1}{d} \qquad d = \frac{T}{N} \quad (8)$$

where T represents gesture path length and N refers to number of states. Matrix $B$ is an $N$-by-$M$ emitted symbols in which $b_{im}$ represents the probability of observation each character symbol in state $i$. Since the states of HMM gesture model discrete, every element in matrix $B$ is sated with the similar value (Eq. 9).

$$b_{im} = \frac{1}{M} \quad (9)$$

where $_{im}$ can carry out the number of discrete symbols and the number of states, respectively.

$$B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1M} \\ b_{21} & b_{22} & \cdots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NM} \end{pmatrix} = \begin{pmatrix} \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} \\ \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} \end{pmatrix} \quad (10)$$

For every time, the HMM state can itself leap back, or first to the close subsequent state. To ensure the beginning from the start state 1, the initial vector $\pi$ should be adjusted as:

$$\pi = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^T \quad (11)$$

Here, the Baum-Welch technique is very resourceful. Time and again a moral model is got after 6-10 iterations. The learned gestures model be supple enough to accurately symbolize a new examination sequence, which never occur throughout learning process. The learning process is iterated till the emission be changed and transition matrices be converged. The convergence is contented when the change be fewer than tolerance $\varepsilon$ =0.001 (Eq. 12), or reaches to the greatest number of iterations.

$$\sum_{i=1}^{N}\sum_{j=1}^{N}|\hat{a}_{ij} - a_{ij}| + \sum_{j=1}^{N}\sum_{m=1}^{M}|\hat{b}_{jm} - b_{jm}| < \epsilon \quad (12)$$

The main reason behind exploiting the tolerance is to manage the required steps number using Baum-Welch function to effectively carry out its principle.

## 6. EXPERIMENTS DISCUSSION

For our proposed system, the hand is segmented and localized color information and 3D depth information with respect to complex scene. For that purpose, the Gaussian Mixture Model (GMM) with $YC_bC_r$ is employed to automatically detect skin region of hands and face [27]. The chrominance channels ($C_b$, $C_r$) are used since they contain only color information and the outcome of brightness disparity is decreased by disregarding $Y$ channel.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This is done based on the construction of a large dataset which contains skin and non-skin pixels. The GMM models the skin pixels in conjunction with k-mean clustering algorithm which initializes the required parameters. Thereby, the hand mean-shift procedure is done to track and generate the gesture path from fingertip detection at each frame. To increase the density estimation, Epanechnikov kernel uses the histogram of hand color to consider a small weight from the center [28]. Then the Bhattacharyya coefficient [18] is used as a similarity measure to influence the facade between the target hand and the candidate hand by maximizing Baye's error. So, Bhattacharyya coefficient gives best match for hand's target sequentially. It is noted that, the mean depth value of the Region of Interest (ROI) in a current frame easily resolve the overlap issue between hand and face. Consequentially, the mean shift vector is recalculated to carry out the optimality in finding recursively further fingertip position. Moreover, the Kalman iteration is called to drive and predicate the position of hand target.

*Figure 8: Temporal Evolution of Three Higher Priorities of Gestures Probabilities. At t = 24, The Higher Priority to Gesture '2' but at t = 40 The Higher Priority Refers to Gesture '8' and at t = 53 The Final Result with Gesture '8'.*

The isolated hand gesture application for alphabet (A-Z) and number (0-9) was considered to carry out the affection of real-time recognition to control IoT products. Bumblebee camera system to capture stereo videos sequences is used at 25FPS with 240 × 320 image resolution. The projected system was carried out using Matlab language and C++ language. The HMM classifier is developed to learn and test the gesture path generated by fingertip. Our experiment showed that each gesture of alphabet character A-Z or number 0-9 was relied on 30 videos samples such that ten video samples for testing and another twenty video samples for learning. Formally speaking, the dataset contains 720 image sequences for learning and 360 image sequences to inference gestures. In Figure 8, the output is to gesture path '8', in which the gesture path is denoted by red color. To evaluate the proposed system, we depend on the following criteria; suppose that t is the number of testing data samples (i.e., number of testing data that has the tenth for each hand gesture path), in which the recognition of valid gesture is denoted by $v_j$ and invalid gesture path is to $\overline{v}_j$ (Eq. 13).

$$t = v_j + \overline{v}_j \ , \qquad j = 1,2,...,36 \tag{13}$$

Here, the index $j$ represents a gesture path of alphabet characters or number. Eq. 14 computes the valid recognition percentage of every hand gesture path whereas the total percentage of inferencing all tested gesture paths are given by Eq. 15.

$$h_j = \frac{v_j}{t}.100 \tag{14}$$

$$\Omega = \frac{1}{36}\sum_{j=1}^{36}\Gamma_j \tag{15}$$

Where $\Gamma_j$ refers to the result of each alphabet character or number, but $\Omega$ is the total recognition value.

Similarly, the recognition on learning dataset is evaluated with respect to every gesture number and character. Table 1, illustrates the rate of overall recognition which represents the average of learning as well as testing samples result. It is being noted that, the overall recognition rate achieved using $F_c$+$F_{sc}$ polar features is 95.84%. Additionally, our experiment yields capable results and realized 98.61% and 93.06% classification rate for training and testing database, respectively. In other words, unlike previous work such as [7], our approach is able to recognize hand gesture even if the background has colors similar to the face and hands. This is because of GMM which is learnt using skin and non-skin datasets.

*Table 1: Results of Gestures Recognition*

| *Feature Type* | *Dataset* | | *Recognition result (%)* | | |
|---|---|---|---|---|---|
| | Train | Test | Train | Test | Overall |
| $F_c$ | 720 | 360 | 94.42 | 84.73 | 89.58 |
| $F_{sc}$ | 720 | 360 | 95.83 | 86.11 | 90.75 |
| $F_c$+$F_{sc}$ | 720 | 360 | 98.61 | 93.06 | 95.84 |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

## 7. CONCLUSION

One of the main motivations of IoT is to provide users with the capability to control and monitor things (i.e., devices) using Internet technologies. Gesture control can help IoT services' users to have a better experience when controlling the IoT products. However, not much attention has been given by researchers for the gesture control on IoT systems. In this paper, we propose a gesture recognition system for gesture control on IoT services. In particular, the gesture recognition system is proposed to classify the hand fingertip trajectory using Hidden Markov Model (HMM). Firstly, the Gaussian Mixture Model (GMM), color information with 3D depth information segment and localize the position of hands. Secondly, mean-shift procedure with kalman filter is used to track fingertip to produce the gesture path. Dynamic features is extracted with respect to polar coordinates and then employed for HMM using k-mean clustering technique. Finally, Baum-welch algorithm is used to carry out the learning process and the viterbi algorithm is to recognize the gesture. Our experiment provides a promising result where the average recognition rates achieved 98.61% and 93.06% for training and testing dataset, respectively. Our approach is able to recognize hand gesture even if the background has colors similar to the face and hands. This is because of GMM which is learnt using skin and non-skin datasets.

For future work, we will investigate the discriminative classifier to deal with words and sentences to make the system more realistic in controlling IoT products.

## ACKNOWLEDGEMENTS

My sincere thanks and appreciation to Dr. Mahmoud Elmezain, Tanta University, Egypt for direct support towards providing us with the gesture database, which helped us to address our research work.

## REFRENCES:

[1] Perera, C., Liu, C.H., Jayawardena, S. and Chen, M., A Survey on Internet of Things From Industrial Market Perspective, *IEEE Access*, Vol. 2, 2014, pp.1660-1679.

[2] Whitmore, A., Agarwal, A. and Da Xu, L., The Internet of Things - A Survey of Topics and Trends, *Information Systems Frontiers*, Vol. 17, No. 2, 2015, pp.261-274.

[3] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M., Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, *IEEE Communications Surveys & Tutorials*, Vol. 17 No. 4, 2015, pp.2347-2376.

[4] Bowden, R., Zisserman, A., Kadir, T, and Brady, M., Vision Based Interpretation of Natural Sign Languages. *In Proceedings of the International Conference on Computer Vision Systems*, 2003.

[5] Botta, A., De Donato, W., Persico, V. and Pescapé, A., Integration of Cloud Computing and Internet of Things: A Survey, *Future Generation Computer Systems*, Vol. 56, 2016, pp.684-700.

[6] Want, R., Schilit, B.N. and Jenson, S., Enabling the Internet of Things, *IEEE Computer*, Vol. 48, No. 1, 2015, pp.28-35.

[7] Han, X. and Rashid, M.A., Gesture and Voice Control of Internet of Things. *In Proceedings of IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, 2016, pp.1791-1795.

[8] Hussain, M., *Automatic* Recognition of Sign Language Gestures, *Master's Thesis*, Jordan University of Science and Technology, 1999.

[9] Handouyahia, M., Ziou, D. and Wang, S., Sign Language Recognition Using Moment-based Size Functions. *In Proceedings International Conference on Vision Interface*, 1999, pp.210-216.

[10] Malassiotis, S. and Strintzis, M., Real-time Hand Posture Recognition Using Range Data, *Image and Vision Computing*, Vol. 26, No. 7, 2008, pp.1027-1037.

[11] Licsar, A., and Sziranyi, T., Supervised Training Based Hand Gesture Recognition System. *In Proceedings of the International Conference on Pattern Recognition*, 2002, pp.999-1002.

[12] Freeman, W., and Roth, M., Orientation Histograms for Hand Gesture Recognition" *In Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 1994, pp.296-301.

[13] Yoon, H., Soh, J., Bae, Y.J., Yang, H.S., Hand Gesture Recognition Using Combined Features of Location, Angle and Velocity, *Journal of Pattern Recognition*, Vol. 34, No. 7, 2001, pp. 1491-1501.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[14] Klette, R., Schluns, K., Koschan, A., *Computer Vision: Three-Dimensional Data from Images*, Springer, Singapore, 1998.

[15] A. Al-Hamadi, O. Rashid, and B. Michaelis, Posture Recognition using Combined Statistical and Geometrical Feature Vectors Based on SVM, *International Journal of Computational Intelligence*, Vol. 6, No. 1, 2010, pp.7-14.

[16] L. Jin, C. Chen, L. Zhen, and J. Huang, Real-Time Fingertip Detection from Cluttered Background for Vision-based HCI, *Journal of Communication and Computer*, Vol. 2, No. 9, 2005, pp.1-8.

[17] Davis, J., Bradski, G., Real-time Motion Template Gradients using Intel CVLib, *In Proceeding of IEEE ICCV Workshop on Framerate Vision*, 1999, pp.1-20 Year of Publication: 1999).

[18] Khalid, S., Ilyas, U., Sarfaraz, S., Ajaz, A, Bhattacharyya Coefficient in Correlation of Gary-Scale Objects, (2006) *Journal of Multimedia*, Vol. 1, No. 1, pp.56–61.

[19] Elmezain, M., Al-Hamadi, A., Michaelis, B., Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences, *The Journal of WSCG'08*, Vol. 16, No. 1, 2008, pp.65–72.

[20] Niese, R., Al-Hamadi, A., Michaelis, B., A Novel Method for 3D Face Detection and Normalization, *The Journal of Multimedia*, Vol. 2, No. 5), pp.1–12.

[21] Comaniciu, D., Ramesh, V., Meer, P, Kernel-Based Object Tracking, *IEEE Transactions PAMI*, Vol. 25, No. 5, 2003, pp.564-577.

[22] Ho-Sub, Y., Jung, S., Young, B., Hyun S., Hand Gesture Recognition using Combined Features of Location, Angle and Velocity, *Journal of Pattern Recognition*, Vol. 34, No. 7, 2001, pp.1491-1501.

[23] A. Al-Hamadi, M. Elmezain, and B. Michaelis, Hand Gesture Recognition Based on Combined Features Extraction, *International Journal of Information Technology*, Vol. 6, No, 1, 2010, pp.1-6.

[24] L. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *In Proceedings of the IEEE*, Vol. 77, No. 2, 1989, pp.257-286.

[25] I. V. Tetko, D. J. Livingstone, and A. I. Luik, Neural Network Studies. Comparison of Over_Fitting and Overtraining, *Journal of Chemical Information and Computer Sciences*, Vol. 35, No. 5, 1995, pp.826-833.

[26] M. Elmezain, A. Al-Hamadi, and B. Michaelis, Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences, *Journal of WSCG*, Vol. 16, No. 1, 2008, pp.65-72.

[27] Elmezain, M., Al-Hamadi, A., Appenrodt, J., and Michaelis, B., A Hidden Markov Model-Based Continuous Gesture Recognition System for Hand Motion Trajectory. *In Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2008, pp.519-522.

[28] Comaniciu, D., Ramesh, S., and Meer, P., Kernel-Based Object Tracking, *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 25, No. 5, 2003, pp.564-577.