Spatio-Temporal Feature Extraction-Based Hand Gesture Recognition for Isolated American Sign Language and Arabic Numbers

Mahmoud Elmezain, Ayoub Al-Hamadi, Saira Saleem Pathan, Bernd Michaelis
Institute for Electronics, Signal Processing and Communications (IESK)
Otto-von-Guericke-University Magdeburg, Germany
{Mahmoud.Elmezain, Ayoub.Al-Hamadi}@ovgu.de

Abstract

This paper proposes a system to recognize isolated American Sign Language and Arabic numbers in real-time from stereo color image sequences using Hidden Markov Models (HMMs). Our system is based on three main stages; preprocessing, feature extraction and classification. In preprocessing stage, color and 3D depth map are used to detect and track the hand. The second stage, 3D combined features of location, orientation and velocity with respect to Cartesian and Polar systems are used. Additionally, k-means clustering is employed for HMMs code-word. In the final stage, the hand gesture path is recognized using Left-Right Banded topology (LRB) in conjunction Viterbi path. Experimental results demonstrate that, our system can successfully recognize isolated hand gestures with 98.33% recognition rate.

1. Introduction

The use of hand gesture is an active area of research in the vision community, mainly for the purpose of sign language recognition and Human Computer Interaction (HCI). A gesture is spatio-temporal pattern, which may be static or dynamic or both. Static morphs of the hands are called postures and hand movements are called gestures. The goal of gesture interpretation is to push the advanced human-machine communication to bring the performance of human-machine interaction close to human-human interaction. In the last decade, several methods of potential applications [2, 4, 6] in the advanced gesture interfaces for HCI have been suggested but these differ from one to another in their models. Some of these models are Neural Network [2], HMMs [4] and Fuzzy Systems [6].

Vassilia et al. [14] developed a system that could recognize both isolated and continuous Greek Sign Language (GSL) sentences where the orientation vector is extracted from images and then used in sentences as input to HMMs. Ho-Sub et al. [5] introduced a hand gesture recognition method, which used the combined features of location, angle and velocity to determine the discrete vector that is used as input to HMMs. This method runs over the alphabets (A-Z), numbers (0-9), six edit commands and six drawing elements. Nianjun et al. [11] proposed a method to recognize the 26 letters from A to Z by using different HMMs topologies with different states number. But, these methods run off-line over a non complex background. Nguyen et al. [10] developed a hand gesture recognition system to recognize real-time gesture in unconstrained environments where the system was tested to a vocabulary of 36 gestures including the American Sign Language letter spelling alphabets and digits. But, this method [10] studies the posture of the hand, not the hand motion trajectory as it is in our system. One of such problems, which arise in real-time hand gesture recognition, is to caused by the fact that the same gesture varies in shape, trajectory and duration, even for the same person. So, HMMs is used in our system where it is capable of modeling spatio-temporal time series.

The main contribution of this paper is to examine the capabilities of combined features of location, orientation and velocity for gesture recognition, which are obtained from spatio-temporal hand gesture path. Additionally, it proposes a real-time capable system that recognize the alphabets characters (A-Z) and numbers (0-9) from stereo color image sequences by the motion trajectory of a single hand using HMMs. Color and 3D depth map are used to detect hands. Furthermore, the hand trajectory is estimated using Mean-shift algorithm [1] and Kalman filter [15] in conjunction with 3D depth map. The system is tested on a different experiments with varying features that are extracted from Cartesian and Polar systems to decide which feature is the best in terms of result. Each alphabet and each number is based on 30 video (20 for training and 10 for testing). The recognition rate that achieved on testing gestures is 98.33%. The rest of this paper is organized as follow; Section 2 demonstrates the suggested system in three sub-sections. Experimental results are described in Section 3. Finally, Section 4 ends with a few conclusion.

2. Gesture Recognition System

We propose an automatic system that recognizes isolated gesture for Alphabets (A-Z) and Arabic numbers (0-9) in
real-time from stereo color image sequences by the motion trajectory of a single hand using HMMs. In particular, the proposed system consists of three main stages; an automatic hand segmentation and tracking, feature extraction and classification (Fig. 1).

![Figure 1. Simplified structure showing the main computational modules for isolated gesture recognition system.](image)

1. Preprocessing; localize and track the hand to generate its motion trajectory (gesture path)
2. Feature extraction; Clustering extracted features to generate discrete vectors, which are used as input to HMMs recognizer.
3. Classification; the gesture path is recognized using discrete vector and Left-Right Banded topology.

The hand graphical gesture consists of 26 alphabet characters from A to Z and 10 Arabic numbers from 0 to 9 where the gesture shapes are shown in Fig. 2.

![Figure 2. Hand gesture path shapes for Alphabets character (A-Z) and Numbers (0-9).](image)

### 2.1. Automatic Hand Segmentation and Tracking

A method for detection and segmentation of the hands in stereo color images with complex background is described where the hand segmentation and tracking takes places using 3D depth map and color information. This stage contains two steps; skin segmentation by using Gaussian Mixture Model (GMM) over \( YC_bC_r \) color space [13], and hand tracking by Mean-shift algorithm in conjunction with Kalman filter and depth information.

### 2.1.1 Hand Segmentation and Localization

Segmentation of skin colored regions becomes robust if only the chrominance is used in analysis. Therefore, \( YC_bC_r \) color space is used in our system where \( Y \) channel represents brightness and \( (C_b, C_r) \) channels refer to chrominance. We ignore \( Y \) channel to reduce the effect of brightness variation and use only the chrominance channels that fully represent the color information. A large database of skin and non-skin pixels is used to train the Gaussian model. The Gaussian Mixture Model begins with modeling of skin using skin database where a variant of \( k \)-means clustering algorithm [3], [5], [7] performs the model training to determine the initial configuration of GMM parameters. For the skin segmentation of hands and face in stereo color image sequences an algorithm is used, which calculates the depth value in addition to skin color information. The depth information (Fig. 3(c)) solves the overlapping problem between hands and face where it is obtained by passive stereo measuring based on cross correlation and the known calibration data of the cameras. Several clusters are composed of the resulting 3D-points. The clustering algorithm can be considered as kind of region growing in 3D that used two criteria; skin color and Euclidean distance. Furthermore, this method is more robust to the disadvantageous lighting and partial occlusion, which occur in real-time environment. In addition, blob analysis is used to derive the boundary area, bounding box and hands centroid point. For more details, the reader can refers to [4], [12].

![Figure 3. (a) Left image frame of video stream. (b) Right image. (c) The depth value of left and right image via the Bumblebee stereo camera.](image)

### 2.1.2 Hand Tracking

After localization of the hand’s target from the segmentation step, we find its color histogram with Epanechnikov kernel [1]. This kernel assigns smaller weights to pixels farther from the center to increases the robustness of the density estimation. To find the best match of our hand target in the sequential frames, the Bhattacharyya coefficient [8] is used to measure the similarity by maximizing Bayes error that arising from the comparison of the hand target and candidate. We take in our consideration the mean depth value that is computed from the previous frame for the hand region to solve the overlapping between hands and face. The mean-shift procedure is defined recursively and performs the optimization to compute the mean shift vector. After
each mean-shift optimization that gives the measured location of the hand target, the uncertainty of the estimate can also be computed and then followed by the Kalman iteration, which drives the predicted position of the hand target. Thereby, the hand gesture path is obtained by taking the correspondences of detected hand between successive image frames (Fig. 4(e) & (f)). For more details, the reader can refer to [1], [4], [12].

2.2. Feature Extraction

There is no doubt that selecting good features to recognize the hand gesture path plays significant role in system performance. There are three basic features; location, orientation and velocity. We analyze the effectiveness of these features that are extracted from a hand trajectory and also combine them to test their recognition rate. The 3D dynamic features are classified in two categories; features in Cartesian space \((x, y)\) and features in Polar space \((\rho, \varphi)\).

2.2.1 Features in Cartesian Space

A gesture path is spatio-temporal pattern that consists of hand centroid points \((x_{hand}, y_{hand})\). The coordinates in the Cartesian space can be extracted from gesture frames directly. We consider two types of location features. The first location feature is \(Lc\) that measures the distance from the centroid point to all points of the hand gesture path because different location features are generated for the same gesture according to different starting points (Eq. 1). The second location feature is \(Lsc\), which is computed from the start point to the current point of hand gesture path (Eq. 3).

\[
Lc_t = \sqrt{(x_{t+1} - x_c)^2 + (y_{t+1} - y_c)^2} \tag{1}
\]

\[
(C_x, C_y) = \frac{1}{n} \sum_{t=1}^{n} (x_t, y_t) \tag{2}
\]

\[
Lsc_t = \sqrt{(x_{t+1} - x_1)^2 + (y_{t+1} - y_1)^2} \tag{3}
\]

where \(t = 1, 2, ..., T - 1\) and \(T\) represents the length of hand gesture path. \((C_x, C_y)\) refers to the centroid of gravity at the \(n\) points. To verify the real-time implementation, the centroid point of gesture path is computed after each frame.

The second basic feature is the orientation, which gives the direction along which the hand when traverses in space during the gesture making process. As described above, the orientation feature is based on the calculation of the hand displacement vector at every point and is represented by the orientation according to the centroid of gesture path (\(\theta_{1t}\)), the orientation between two consecutive points (\(\theta_{2t}\)) and the orientation between start and current gesture point (\(\theta_{3t}\)).

\[
\theta_{1t} = \arctan \left( \frac{y_{t+1} - y_c}{x_{t+1} - x_c} \right) \tag{4}
\]

\[
\theta_{2t} = \arctan \left( \frac{y_{t+1} - y_t}{x_{t+1} - x_t} \right) \tag{5}
\]

\[
\theta_{3t} = \arctan \left( \frac{y_{t+1} - y_1}{x_{t+1} - x_1} \right) \tag{6}
\]

The third basic feature is the velocity, which plays an important role during gesture recognition phase particularly at some critical situations. The velocity is based on the fact that each gesture is made at different speeds where the velocity of the hand decreases at the corner point of a gesture path. The velocity is calculated as the Euclidean distance between the two successive points divided by the time in terms of the number of video frames as follows;

\[
V_t = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \tag{7}
\]

In the Cartesian coordinate system, we use different combination of features to obtain a variety feature vectors. For example, the feature vector at frame \(t + 1\) is obtained by union of locations features \((Lc_t, Lsc_t)\), locations features with velocity feature \((Lc_t, Lsc_t, V_t)\), orientations features \((\theta_{1t}, \theta_{2t}, \theta_{3t})\), orientations features with velocity feature \((\theta_{1t}, \theta_{2t}, \theta_{3t}, V_t)\) and locations features with orientations features and velocity feature \((Lc_t, Lsc_t, \theta_{1t}, \theta_{2t}, \theta_{3t}, V_t)\).

Each frame contains a set of feature vectors at time \(t\) where the dimension of space is proportional to the size of feature vectors. In this manner, gesture is represented as an ordered sequence of feature vectors, which are projected and clustered in space dimension to obtain discrete code-word that are used as an input to HMMs. This is done using k-means clustering algorithm [3], [7], which classifies the gesture pattern into \(K\) clusters in the feature space.

2.2.2 Features in Polar Space

Polar coordinate can be directly calculated from the Cartesian coordinates that are generated from gesture images. To obtain the normalized polar coordinates, we use the radius from the center point of the gesture path (Eq. 9) and the radius between the start and the current gesture point (Eq.11).

\[
r_{c_{\text{max}}} = \max(Lc_t), \quad \rho_{ct} = \frac{Lc_t}{r_{c_{\text{max}}}}, \quad \varphi_{ct} = \frac{\theta_{1t}}{2\pi} \tag{8}
\]

\[
F_c = \{(\rho_{c1}, \varphi_{c1}), (\rho_{c2}, \varphi_{c2}), ..., (\rho_{cT-1}, \varphi_{cT-1})\} \tag{9}
\]

\[
r_{sc_{\text{max}}} = \max(Lsc_t), \quad \rho_{sc} = \frac{Lsc_t}{r_{sc_{\text{max}}}}, \quad \varphi_{sc} = \frac{\theta_{2t}}{2\pi} \tag{10}
\]

\[
F_{sc} = \{(\rho_{sc1}, \varphi_{sc1}), (\rho_{sc2}, \varphi_{sc2}), ..., (\rho_{scT-1}, \varphi_{scT-1})\} \tag{11}
\]

where \(r_{c_{\text{max}}}\) is the longest distance from the center point of each hand trajectory at frame \(t + 1\) and \(r_{sc_{\text{max}}}\) represents the longest distance from the start point to each point in the hand gesture path.

In the Polar space, we use different combination of features to obtain a variety feature vectors. For example, the feature vector at frame \(t + 1\) is obtained by union of locations features from the centroid point with velocity feature \((\rho_{ct}, \varphi_{ct}, V_t)\), locations features from the start and the current point with velocity feature \((\rho_{sc}, \varphi_{sc}, \rho_{ct}, \varphi_{ct}, V_t)\), and a combination of all \((\rho_{ct}, \varphi_{ct}, \rho_{sc}, \varphi_{sc}, V_t)\).
2.2.3 Vector Quantization

The extracted features are quantized to obtain the discrete symbols. When the basic features such as locations and velocity are used separately, these features are normalized and multiplied by a different scalar ranging from 10 to 30. On the other side, the normalization of the orientation features is to divide by 10°, 20°, 30° and 40° to obtain its codeword. In addition to the combination features in the Cartesian and the Polar coordinate system, we use k-mean clustering algorithm to classify the gesture feature into K clusters on the feature space. This algorithm is based on the minimum distance between the center of each cluster and the feature point. We divide the set of feature vectors into set of clusters. This allows us to model the hand trajectory in the feature space by one cluster. The calculated cluster index is used as input (i.e. observation symbol) to the HMMs. Furthermore, we usually do not know the best number of clusters in the data set. In order to specify the number of clusters K for each execution of the k-means algorithm, we considered K = 28, 29, ..., 37, which is based on the numbers of segmented parts in all alphabets character (A-Z) and numbers (0-9) where each straight-line segment is classified into single cluster.

Suppose we have n sample of trained feature vectors \(x_1, x_2, ..., x_n\) all from the same class, and we know that they fall into k compact clusters, \(k < n\). Let \(m_i\) be the mean of the vectors in cluster i. If the clusters are well separated, a minimum distance classifier is used to separate them. That is, we can say that \(x\) is in cluster \(i\) if \(\|x - m_i\|\) is the minimum of all the \(k\) distances. The following procedure shows that how to find the \(k\)-means;

- Build up randomly an initial Vector Quantization Codebook for the means \(m_1, m_2, ..., m_k\);
- Until there are no changes in any mean
  - Use the estimated means to classify each sample of train vectors into one of the clusters \(m_i\)
    - for \(i = 1\) to \(k\)
      - Replace \(m_i\) with the mean of all of the samples of trained vector for cluster \(i\)
    - end (for)
  - end (Until)

A general observation is that different gestures have different trajectories in the cluster space, while the same gesture show very similar trajectories.

2.3. Classification

Markov model is a mathematical model of stochastic process, which generate random sequences of outcomes according to certain probabilities [4], [9]. A stochastic process is a sequence of features codewords, the outcomes being the classification of hand gesture path. Evaluation, Decoding and Training are the main problems of HMMs and they can be solved by using Forward-Backward, Viterbi and Baum-Welch (BW) algorithms respectively. Also, HMMs has three topologies; Fully Connected (i.e. Ergodic model) in which any state can be reached from other states, Left-Right (LR) model such that each state can go back to itself or to the following states and Left-Right Banded model where each state can go back to itself or the next state only.

Each reference pattern in the gesture database for alphabets (A-Z) and Arabic numbers (0-9) is modeled by Left-Right Banded model with varying number of states ranging from 3 to 6 states based on its complexity. As, the excessive number of states can generate the over-fitting problem if the number of training samples is insufficient compared to the model parameters. The hand gesture path is classified by selecting the maximal observation probability of gestures model. The maximal gesture model is the gesture whose observation probability is the largest among all 36 gestures (A-Z & 0-9). The type of observed gesture \((O)\) is decided by Viterbi algorithm frame by frame (i.e. accumulatively until it receives the gesture end signal). The following steps show how the Viterbi algorithm works on gesture model \(\lambda\) \((a^g, b^g, \Pi^g);\)

1. Initialization: for \(1 \leq i \leq N\),
   - \(\delta^g_1(i) = \Pi^g_i b^g(o_1)\)
2. Recursion (accumulative observation probability computation): for \(2 \leq t \leq T, 1 \leq j \leq N\),
   - \(\delta^g_t(j) = \max_i[\delta^g_{t-1}(i) a^g_{ij}] b^g(o_t)\)
3. Termination:
   - \(P(O|\lambda) = \max_i[\delta^g_T(i)]\)

where \(N\) is the number of states, \(\Pi^g_i\) represents the initial value for the state \(i\), \(a^g_{ij}\) is the transition probability from state \(i\) to state \(j\), \(b^g(o_t)\) refers to the probability of emitting \(o\) at time \(t\) in state \(j\), and \(\delta^g_t(j)\) represents the maximum likelihood value in state \(j\) at time \(t\).

3. Experimental Results

Our proposed system was capable for real-time implementation and showed good results to recognize alphabets character and Arabic numbers from stereo color image sequences via the motion trajectory of a single hand using HMMs. The input images were captured by Bumblebee stereo camera system that has 6 mm focal length at 15FPS with 240 \(\times\) 320 pixels image resolution, Matlab implementation. In our experimental results, each isolated gesture was based on 30 video sequences, which 20 video samples for training by BW algorithm and 10 video samples for testing (Totally, our database contains 720 video sample for training and 360 video sample for testing). The gesture recognition module match the hand gesture path against the database of reference gestures, to classify which class it belongs to. The higher priority was computed by Viterbi algorithm to recognize the alphabets and numbers in real-time frame by frame over LRB topology with different number of states ranging from 3 to 6. We test the importance of the three basic features (location, orientation, velocity) in the
Figure 4. (a) Recognition rate for the number of locations and velocity feature codes (10, 15, 20, 25, 30). (b) Results for three different orientations with varying feature codes number (9, 12, 18, 36). (c) & (d) Recognition rate according to a combined features in Cartesian and Polar system over feature codes number from 28 to 37. (e) The high priority is alphabet 'P' at t=45 and at t=70 the result is 'R'. (f) Solving overlap problem between hand and face at t=19 and the high priority is '7' at t=27.

Cartesian and the Polar coordinate. Moreover, the observation sequence for HMMs is quantified either by normalization in case of separated features or by the $k$-means clustering algorithm in case of combined features. From Table 1, the recognition ratio of isolated gestures achieved best results with 98.33% using $(L_c, L_{sc}, \theta_1, \theta_2, \theta_3, V)$ feature. The recognition ratio is the number of correctly recognized gestures to the number of tested gestures (Eq. 12).

$$Reco.\ ratio = \frac{\# \text{ recognized gestures}}{\# \text{ test gestures}} \times 100\% \quad (12)$$

According to the separated features in Fig. 4(a) & (b), the orientation features $(\theta_1, \theta_2, \theta_3)$ are better than the recognition rate of the locations $(L_c, L_{sc})$ or the velocity $(V)$ features. This in turn leads to the orientation feature $(\theta_1 = 96.94\%)$ is the most effective among the three basic features. Furthermore, the velocity feature shows a lower discrimination power (57.22%) than that of the orientation features. Also, the $L_{sc}$ feature result is the lowest recognition rate of 32.78%. The testing results from the union features show that the combined features in Polar system yield a higher recognition ratio than the combined features in Cartesian system (Table 1). Additionally, the $(L_c, L_{sc}, V)$, $(\theta_1, \theta_2, \theta_3, V)$ and $(L_c, L_{sc}, \theta_1, \theta_2, \theta_3, V)$ features, which include the velocity information show higher recognition than when using the velocity information (Fig. 4(c)). But the opposite happened in the case of the Polar system (Fig. 4(d)). In short, Fig. 4 show the results of the experiments that were performed to determine the best feature codes number (the best number of feature code is 33 for the feature $(L_c, L_{sc}, \theta_1, \theta_2, \theta_3, V)$). Fig. 4(e) & (f) shows the output of the system for isolated gesture alphabet 'R' and Arabic number '7' respectively, in addition to the solved overlapping problem between hand and face by 3D depth map.

4. Conclusion and Future Work

This paper proposes a system to recognize the American Sign Language alphabets character (A - Z) and Arabic numbers (0 - 9) from stereo color image sequences by the motion trajectory of a single hand using HMMs. This system uses
the combined features of location, orientation and velocity for Cartesian and Polar systems. We have shown that the effective of these features can yield reasonable recognition rates. The database contains 720 video samples for training and 360 video sequences for testing the isolated gestures. The results show that; the proposed system is suitable for real-time application and can successfully recognize isolated gestures with 98.33% recognition rate. The future research will address the hand gesture spotting and recognition for a sentence using fingertips instead of the hand centroid point in conjunction with multi-camera system.

Acknowledgments.

This work is supported by Forschungspraemie (BMBF-Frderung, FKZ 03FPB00213) and Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG).

References


<table>
<thead>
<tr>
<th>Feature type</th>
<th>Feature</th>
<th># Feature code</th>
<th>Training data</th>
<th>Isolated gestures results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separated in Cartesian coordinates</td>
<td>Lc</td>
<td>20</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>Lsc</td>
<td>25</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>θ1</td>
<td>25</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>θ2</td>
<td>18 ; 36</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>θ3</td>
<td>18</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>18</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td>Union in Cartesian coordinates</td>
<td>(Lc, Lsc)</td>
<td>35</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>(Lc, Lsc, V)</td>
<td>33</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>(θ1, θ2, θ3)</td>
<td>33</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>(θ1, θ2, θ3, V)</td>
<td>34</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td>Union in Polar coordinates</td>
<td>(ρc, ϕc)</td>
<td>28</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>(ρsc, ϕsc)</td>
<td>33</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>(ρc, ϕc, V)</td>
<td>31</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>(ρsc, ϕsc, V)</td>
<td>30</td>
<td>720</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>(ρc, ϕc, ρsc, ϕsc, V)</td>
<td>29 ; 30</td>
<td>720</td>
<td>360</td>
</tr>
</tbody>
</table>