A Framework for Hand Gesture Recognition and Spotting Using Sub-gesture Modeling

Manavender R. Malgireddy
University at Buffalo
mrm42@buffalo.edu

Jason J. Corso
University at Buffalo
jcorso@buffalo.edu

Srirangaraj Setlur
University at Buffalo
setlur@buffalo.edu

Venu Govindaraju
University at Buffalo
govind@buffalo.edu

Dinesh Mandalapu
HP Labs,India
dinesh.mandalapu@hp.com

Abstract

Hand gesture interpretation is an open research problem in Human Computer Interaction (HCI), which involves locating gesture boundaries (Gesture Spotting) in a continuous video sequence and recognizing the gesture. Existing techniques model each gesture as a temporal sequence of visual features extracted from individual frames which is not efficient due to the large variability of frames at different timestamps. In this paper, we propose a new sub-gesture modeling approach which represents each gesture as a sequence of fixed sub-gestures (a group of consecutive frames with locally coherent context) and provides a robust modeling of the visual features. We further extend this approach to the task of gesture spotting where the gesture boundaries are identified using a filler model and gesture-completion model. Experimental results show that the proposed method outperforms state-of-the-art Hidden Conditional Random Fields (HCRF) based methods and baseline gesture spotting techniques.

1. Introduction

Gestures are one of the most important modes of communicating with computer in an interactive environment. Recent advances in computer vision and machine learning have led to a number of techniques for modeling gestures in real-time environment [1, 10, 8, 9]. Modeling each video sequence using a single gesture model usually leads to lower performance owing to a high temporal variability. To address this issue, we propose to model each gesture as a sequence of smaller sub-gesture units. Our approach is inspired by recent study in speech and handwriting recognition. In speech recognition, a phoneme is defined as smallest segmental unit employed to form an utterance (speech vector). Similarly in handwriting recognition, a word can be represented as a sequence of strokes, where each stroke is the smallest segmental unit. This motivates our definition of gesture in a natural and intuitive manner, where each gesture can be represented as a sequence of contiguous sub-gestures denoting a consistent action performed by the user. Typical examples of such sub-gestures would be moving both hands from rest to horizontal positions, moving both hands away from each other and resting both hands which if performed in sequence would model a ”zooming in” gesture (see fig 1).

A detailed survey of human movement and gesture modeling can be found in [3]. Most of the earlier techniques on gesture recognition were based on one-vs-all models, where a separate model is trained for each gesture e.g. HMMs and its variants [2, 10, 8]. Recent advances in gesture recognition research suggest that multiclass (one model for all gesture classes) models like HCRF [9] which are considered to be state of art for gesture recognition, outperforms the one-vs-all models since they jointly learn the best discriminative structure among gestures. HCRF provides us an excellent framework for modeling each gesture as a combination of sub-gestures by sharing hidden states among multiple gesture classes. However, this sharing is usually implicit and there is no way to explicitly define a sub-gesture sequence for a given gesture which might be useful for continuous gesture recognition. Moreover, HCRF training algorithms are computationally very expensive as compared to HMMs. To address these issues, we propose a novel variant of HMM for gesture recognition which combines the advantages of HCRF and HMM. Our proposed model explicitly takes the sequence of sub-gestures for a given gesture (gesture

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.921

grammar) and uses it to learn a model for each gesture. This provides us greater flexibility in modeling gestures as the observations (image features) are used to model a smaller sequence of actions (sub-gestures) as compared to a larger sequence (gesture) in the previous methods. Moreover, it also provides a simple framework for explicit sub-gesture modeling and is computationally efficient. Also, the proposed approach does not require exact segmentation of actions in a gesture.

The second advantage of our proposed approach is that it can be easily extended to other gesture related tasks such as gesture spotting. Gesture spotting is an alternative technique to gesture recognition in real time scenarios, where the goal is to locate the gesture boundaries (start and end of a gesture) as well as gesture label [7, 6, 4, 1, 5]. Most of the spotting frameworks rely on learning a threshold or using heuristic rules to detect the start and end frames of a gesture. Our proposed framework addresses this limitation by using a combination of filler model and gesture-completion model to locate gesture boundaries in a probabilistic framework.

2. Proposed Approach

2.1. Gesture Recognition Framework

The task of gesture recognition is to find the maximum likelihood gesture model corresponding to the input video sequence: $\arg\max_i p(V|\lambda_i)$ where λ_i is the model for i^{th} gesture and V is the video of N frames. Let $\phi = \{\phi_1, \phi_2, ..., \phi_Q\}$ be a set of Q sub-gesture models. Every gesture model in the vocabulary is composed of some sequence of the above mentioned sub-gesture models i.e $\lambda_i = \phi_{S1}\phi_{S2}...\phi_{SK}$ where each $\phi_{Sj} \in \phi$ and $k \geq 1$. The above sequence of sub-gestures in a gesture is known as gesture grammar and is provided by the user. Figure 1 shows an example of zoomin gesture and the corresponding sub-gesture sequence

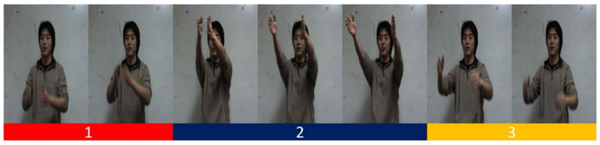

Figure 1. Example of zoomin gesture and corresponding sub-gestures. The first few frames of each sub-gesture in a zoomin gesture is displayed and below each frame is the sub-gesture number

Let S be the segmentation of the video V such that the frames are divided into k continuous segments and the probability $p(V|\lambda_i) = \prod_{i=1}^{k} p(S_i|\phi_{Si})$ is maximized. k is fixed for a given gesture model λ_i and $p(S_i|\phi_{Si})$ is the probability of segment S_i given the sub-gesture model ϕ_{Si}. Such a segmentation S can be found using viterbi decoding. Each sub-gesture model is a left-right HMM and parameter estimation of the sub-gesture models uses similar algorithm to Baum-Welch.

- Initialize the sub-gesture models parameters with the global mean and variance of the data.
- Repeat the following for each of the training video.
- Find $\arg\max_S \prod_{i=1}^{k} p(S_i|\phi_{Si})$ given the gesture.
- Update the parameters of the each sub-gesture model ϕ_{Si} given the segment S_i.

Once the sub-gesture models are learned, construct the gesture models by joining in sequence the sub-gesture models corresponding to a gesture.

2.2. Gesture Spotting Framework

Gesture recognition framework described previously assumes that all the frames in a video belong to a single gesture and the task is to find this gesture. This assumption doesn't hold true in real-time applications, where a continuous stream of video can consist of zero or more gestures. Gesture spotting extends the gesture recognition framework by removing this assumption and defines a video V to consist of p gestures, where $p \geq 0$. The task of gesture spotting framework is to find the boundaries of these p gestures i.e to calculate the pairs $< s_j \; e_j > \forall j$ such that $1 \leq j \leq p$, where s_j and e_j represent the start and end frames of the gesture j respectively.

We solve this problem by building two types of models (filler and gesture-completion) for spotting gestures in a video. The primary motivation here is that motions between gestures can be intuitively thought as some random sequence of sub-gesture and can be modeled using the filler model. In a continuous sequence of frames, if we know that the current frame is the end of a gesture, then *either* all the frames till the current frame belong to the gesture just completed *or* initial few frames belong to a random sequence of sub-gestures and rest of the frames denote a gesture. These two possibilities are modeled by a gesture-completion model by allowing a filler model before a gesture model. A filler model is constructed by joining all the sub-gesture models as follows:

- The observation densities of states remain the same

- The transition probability from entry state (S) of filler model to entry state of sub-gesture models or exit state (E) of filler model is equally likely

$$a_{1j} = \frac{1}{Q+1} \tag{1}$$

  $\forall j \in$ entry state of sub-gesture models and exit state of filler model. Q is the total number of sub-gestures.

- The transition probability from exit state of sub-gesture model to entry state of filler model is 1 (i.e The filler models exit state can only be reached from the entry state of filler model)

Gesture-completion model is constructed similar to the filler model with the only exception that in this case, we add a gesture model (every gesture will have a gesture-completion model) between the start state and exit state of the filler model. Figure 2 shows an example of filler and gesture-completion model in which S and E are the start and exit states of the models respectively.

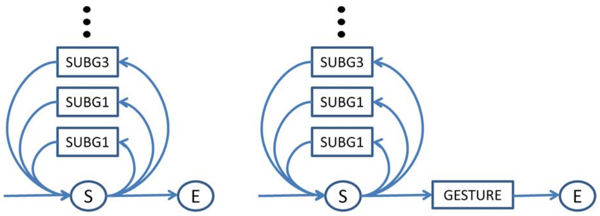

Figure 2. Spotting models (left) filler model (Right) gesture-completion model

Let $p(f_m...f_n|\lambda_{FM})$ be the probability of frames m to n belonging to filler model λ_{FM}. Let $p(f_m...f_n|\lambda_{GCM_i})$ be the probability of frames m to n belonging to i^{th} gesture-completion model λ_{GCM_i}. The pairs $< s_j \ e_j >$ can be found in real time using algorithm 1

3. Experiments and Results

We define five hand gestures for our experiments (see fig 3). Green arrows in the figure shows the motion of hands while performing a gesture. For every two consecutive frames optical flow is extracted and an eight bin histogram is created based on the flow direction. These histograms are used as features in our experiments. Each sub-gesture model is a left-right HMM with eight hidden states and Gaussian densities are used as observations. The number of hidden states used in HCRF model is twelve with window size two. The number of hidden states for the HMM and HCRF models are set by minimizing the error on training data.

Algorithm 1 pseudo code for gesture spotting

```
$m \leftarrow 1$;
$n \leftarrow 0$;
repeat
    Get next frame;
    $n \leftarrow n + 1$;
    if $p(f_m...f_n|\lambda_{FM}) < \max_i p(f_m...f_n|\lambda_{GCM_i})$ then
        $e \leftarrow n$; {e is the end frame of a gesture}
        find start frame s of a gesture by backtracking viterbi path in gesture-completion model;
        $m \leftarrow n$;
    else
        continue;
    end if
until all frames are processed
```

Table 1. Recognition results (% accuracy)

Model	Fold 1	Fold 2	Fold 3	Fold 4
Proposed Method	87.27	80.00	83.64	80.00
HCRF	73.63	77.27	72.72	72.72
HMM	51.82	44.55	51.82	49.09

3.1 Gesture Recognition

Our experiments were conducted on a dataset of five gestures (point (PT), zoomin (ZI), translate (TR), zoomout (ZO) and rotate (RT)) collected from 11 users. The performance of our gesture recognition method is evaluated on a four fold cross validation using 165 videos for training and 110 videos in testing in each fold. As shown in Table 1, our proposed method outperforms both HCRF and conventional HMM based recognition systems.

3.2 Gesture Spotting

Spotting framework is evaluated on two datasets each having four folds created from the recognition dataset. Each fold of spotting dataset is created from corresponding fold of recognition dataset. Each fold has 60 videos for every user. Out of 60 videos of each user, 20 videos are created by randomly selecting and concatenating 3 videos from recognition dataset, 20 videos are created by randomly selecting 5 videos and rest 20 by selecting 7 videos. That yields a total of 660 videos and 3300 gestures for each fold. Dataset-2 is created in a similar manner, but inserting random frames between gestures. The proposed technique is compared with baseline method proposed by Lee and Kim in [7]. Both spotting techniques are evaluated using a com-

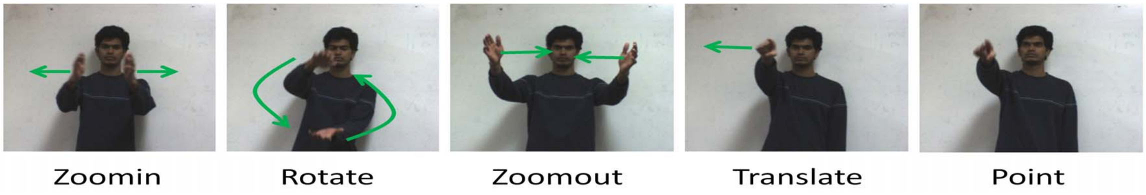

Figure 3. Illustration of five gesture classes

monly known metric for gesture spotting known as ”Detection Rate” (DR) used in [6, 1]. Tables 2 and 3 summarizes the results of spotting framework.

Table 2. Spotting results (% DR) averaged over all folds

Gesture	Method	Overall	
		Dataset I	DataSet II
Point	Proposed / baseline	74.3 / 59.8	73.4 / 50.7
Rotate	Proposed / baseline	67.8 / 67.2	67.2 / 45.7
Translate	Proposed / baseline	74.8 / 70.8	69.9 / 38.4
Zoomin	Proposed / baseline	71.1 / 69.2	68.1 / 38.3
Zoomout	Proposed / baseline	69.7 / 69.7	66.1 / 38.2
Total	Proposed / baseline	71.5 / 66.2	69.2 / 43.1

Table 3. Confusion Matrix for spotting dataset-1 on proposed method. Rows correspond to actual label and, Columns correspond to predicted label.

	PT	RT	TR	ZI	ZO	Deletions
PT	619	4	0	0	0	39
RT	5	507	1	2	0	121
TR	92	3	406	3	18	123
ZI	20	78	19	397	2	155
ZO	56	44	0	16	388	143

4. Conclusion

In this paper, we have proposed a novel sub-gesture modeling technique for gesture recognition and a spotting framework which outperforms state-of-the-art techniques and can also be easily extended to the real-time scenarios. Even though our proposed approach performs better, the low accuracies of the model are due to ample amount of within class variance and less training data. Our current work focuses on extracting more representative features and experimentation on a larger gesture corpus.

References

[1] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, 2008.

[2] M. Assan and K. Grobel. Video-based sign language recognition using hidden markov models. In *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, pages 97–109, London, UK, 1998. Springer-Verlag.

[3] A. F. Bobick, J. W. Davis, I. C. Society, and I. C. Society. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:257–267, 2001.

[4] J. J. Corso, G. Ye, and G. D. Hager. Analysis of composite gestures with a coherent probabilistic graphical model. *Virtual Real.*, 9(1):93–93, 2005.

[5] H. Junker, O. Amft, P. Lukowicz, and G. Tröster. Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recogn.*, 41(6):2010–2024, 2008.

[6] H. Kang, C. W. Lee, and K. Jung. Recognition-based gesture spotting in video games. *Pattern Recogn. Lett.*, 25(15):1701–1714, 2004.

[7] H.-K. Lee and J.-H. Kim. Gesture spotting from continuous hand motion. *Pattern Recogn. Lett.*, 19(5-6):513–520, 1998.

[8] H. Li and M. Greenspan. Multi-scale gesture recognition from time-varying contours. In *ICCV ’05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, pages 236–243, Washington, DC, USA, 2005. IEEE Computer Society.

[9] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1521–1527, 2006.

[10] R. Yang and S. Sarkar. Gesture recognition using hidden markov models from fragmented observations. In *CVPR ’06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 766–773, Washington, DC, USA, 2006. IEEE Computer Society.