# A Decomposition Approach for Complex Gesture Recognition Using DTW and Prefix Tree

# A Decomposition Approach for Complex Gesture Recognition Using DTW and Prefix Tree

Hui Chen, Tarig Ballal, and Tareq Al-Naffouri *
Division of Computer, Electrical and Mathematical Science and Engineering (CEMSE),
King Abdullah University of Science and Technology (KAUST)

## Abstract

Gestures are effective tools for expressing emotions and conveying information to the environment. Sequence matching and machine-learning based algorithm are two main methods to recognize continuous gestures. Machine-learning based recognition systems are not flexible to new gestures because the models have to be trained again. On the other hand, the computational time that matching methods required increases with the complexity and the class of the gestures. In this work, we propose a decomposition approach for complex gesture recognition utilizing DTW and prefix tree. This system can recognize 100 gestures with an accuracy of 97.38%.

**Keywords:** Gesture Recognition, Time Sequence, DTW, Prefix Tree, Human-machine-interaction

**Index Terms:** Human-centered computing—Human computer interaction (HCI)—HCI design and evaluation methods—Gestural input;

## 1 Introduction

Human gestures are indispensable tools for expressing emotions, conveying information to the environment and interacting with the objects in virtual reality environment [2].

Generally, a gesture recognition system consists of motion tracking and classification [1]. The movement of the hand can be captured using cameras, acceleration sensors, electromagnetic and acoustic signals [3]. These technologies are used in distinct scenarios and the systems can be evaluated based on various criteria such as accuracy, resolution, latency, user comfort, cost and so on [2].

Machine learning algorithms and template matching are two mainstreams of gesture recognition. The former ones are trained with labeled gesture sequence and the well-trained models can do classification to the input data in the same data format. However, the models have to be trained with a new set of training data if the user wants to define new gestures. Template matching methods are relatively flexible with newly introduced gestures. This kind of method computes the distance or the similarity between the unclassified sequence and template sequence. Nevertheless, the drawback of template matching methods is the increasing computational complexity of length and classes of the gestures.

In order to create a flexible and fast complex gesture recognition algorithm. We propose a component-filtering (com-filter) classifier utilizing DTW to filter potential hand gesture components and use prefix tree to eliminate wrong gestures. This system is tested on a 100-gesture dataset collected from 8 volunteers and the classification accuracy reaches 97.38%.

## 2 Gesture Modeling

### 2.1 Gesture Model

The gesture model in this work is described as a 2-D time series $[\mathbf{a}, \mathbf{b}]$ containing horizontal and vertical location information. However, the depth information can also be introduced to describe the gesture using a 3-D location vector. We use English letters as gesture components and form a 100-gesture dataset by writing the word (from the most widely-used 100 English words) in one stroke.

### 2.2 Feature Extraction

To deal with the different shapes and performing speed of different users. Instead of using location information directly, we use the orientation sequence $\mathbf{h}$ to represent the hand movement and the feature extraction procedure is shown in Algorithm 1.

**Algorithm 1** Feature Extraction

```
$i, ind \leftarrow 1$
for $j = 2$ to $N$ do
    $ori \leftarrow [b_j - b_i, a_j - a_i]$
    if norm$(ori) < TH_{dis}$ then
        $h_{ind} \leftarrow$ atan2$(ori[1], ori[2])$
        $i \leftarrow j$
        $ind \leftarrow ind + 1$
return $\mathbf{h}$
```

Where atan2$(\cdot)$ is the four-quadrant inverse tangent function which returns the orientation in $(-\pi, \pi]$, $a_i$ and $b_i$ are the $i$-th elements of the horizontal and vertical location sequences of the gesture. The feature extraction procudure can solve different writing style problem and is able to reduce the computational complexity by subsampling and merging adjacent similar orientations if necessary.

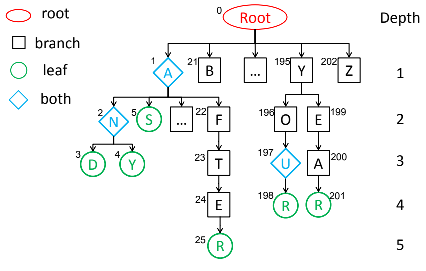

Figure 1: A section of the prefix tree.

### 2.3 Prefix Tree

Prefix tree is widely used in information retrieval applications. Instead of creating all the gesture templates, gestures can be decomposed into a set of gesture components and stored in an interconnected way for fast retrieval as shown in Fig. 1. Each node has a unique index and four types of the nodes are *root*, *branch*, *leaf* and *both* (which means this node is a leaf as well as a branch). This

[*] e-mail: (hui.chen, tarig.ahmed, tareq.alnaffouri)@kaust.edu.sa

tree structure helps the system to identify a gesture with several subgestures.

## 3 Gesture Recognition

### 3.1 Dynamic Time Warping

Dynamic time warping (DTW) [4] is a method calculating the distance between two sequences with varying length. The distance between two features is defined as

$$s_{i,j} = 1 - \cos(h_i - h_j). \tag{1}$$

Then, an accumulated cost matrix $\mathbf{C}$ can be calculated as

$$c_{i,j} = \begin{cases} s_{i,w} & i = 1 \\ \sum_{w=1}^{i} s_{w,j} & j = 1 \\ u_{i,j} + s_{i,j} & otherwise, \end{cases} \tag{2}$$

where $u_{i,j} = \min\{c_{i-1,j-1}, c_{i-1,j}, c_{i,j-1}\}$.

An example of the accumulated cost matrix between the word 'BECAUSE' and the letters 'A','B','C','E' is shown in Fig. 2. The value in the last column indicates the distance between part of the observed sequence (red rectangle area) and the reference letter.

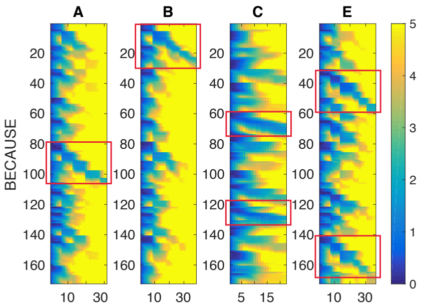

Figure 2: The cost matrices between 'BECAUSE' and several letters.

All candidate letters with average distances smaller than a threshold $TH\_l$ will be filtered into a list $L$ which contains the information of the starting position $Ps = Ps_{i,J}$, ending position $Pe = Pe_{i,J}$ and distance $s = c^m_{i,j}$. This list L, as well as the prefix tree, will be used in com-filter classifier.

### 3.2 Com-Filter Algorithm

The com-filter will utilize the candidate letter list $L$ obtained from W-D-DTW and match them with the prefix tree to find the candidate gesture. This algorithm matches the gesture components depth by depth from the root of the prefix tree to the deepest leaf. After evaluating the candidate components in each depth $w$, the top $c_n$ candidates with the least average distance are kept in the list $L_w$. The list $L_w$ contains all the information of the candidates which are index $Ind$, stop position $Stop$, coverage samples $Cvg$, accumulated distance $Dis$, average distance $Davg$ and finish indicator $F$. The candidates in depth $w+1$ will be all the children nodes of the node in the list $L_w$. The whole Com-Filter algorithm can be performed in several steps:

1. Do DTW and obtain the candidate information list $L$;
2. Initialize the $L_w$ with $w = 0$ and set $L_0.Val$ = 'ROOT' and reset all the other parameters.
3. While loop $[L_{w+1}, depth] =$ update_L$(L_w, depth)$ until all the $L_w.F$ values are 1.
4. The final $L_w$ contains all the possible word candidates and their information.

The procedure update_L$(\cdot)$ is the function that updates the candidates list $L_{w+1}$ in depth $w+1$ as shown in Algorithm 2. $WIN_{max}$ is the window size for the candidate selecting starting area, $c$ is the max candidate number kept to the next depth.

**Algorithm 2** update_L

```
for $i = 0$ to length$(L_w)$ do
    if $L_w(i).F = 0$ then
        $S_{area} \leftarrow [L_w(i).Stop), (L_w(i).Stop + WIN_{max})]$
        $L_c \leftarrow find(L.Ps$ in $S_{area})$
        for $j = 0$ to length$(L_w(i).Ind.Children)$ do
            $L_{new} \leftarrow find(L_w(i).Ind.Children(j)$ in $L_c)$
            Update $L_{new}.all$
Check Validity, Pick top $c$ with least $D_{avg}$ in $L_{new}$ as $L_{w+1}$
$depth \leftarrow depth + 1$
return $L_{w+1}, depth$
```

In 'Update $L_{new}.all$' procedure, all the related values except $L_{new}.F$ will be updated. In 'Check Validity' procedure, $L_{new}.F$ will be set equal to 1 for $leaf$ nodes, set to 0 for $branch$ node, duplicated and set to 1 and 0 for $both$ nodes. In addition, the candidate in $L_{new}$ will be removed if the gesture is incomplete (current node is not a leap) till the end of the processing.

Com-filter algorithm selects $c$ candidates at each depthin a tree structure and hence reduces the computational complexity of classification. In addition, this algorithm provides the convenience for real-time processing and prediction.

## 4 Experiments and Results

We use 100-frequently used English words as complex gesture pool to test the algorithm and only 26 gesture components are used to classify the gesture. A total number of 800 gesture data from 8 volunteers were collected and the recognition results with top-k candidates using Com-Filter algorithms are shown in Table 1. The algorithm realized in Matlab can be found in *MY GITHUB*.

Table 1: Gesture Recognition Results

| Method | 1 candidate | 2 candidates | 3 candidates |
|---|---|---|---|
| Com-Filter | 95.00% | 96.88% | 97.38% |

## 5 Conclusion

In order to create a flexible and fast complex gesture recognition algorithm. We propose a component-filtering (com-filter) classifier utilize DTW to filter potential hand gesture components and use prefix tree to eliminate wrong gestures. This system is tested on a 100-gesture dataset collected from 8 volunteers and the classification accuracy reaches 97.38%.

## References

[1] M. Chen, G. AlRegib, and B.-H. Juang. Air-writing recognitionpart ii: Detection and recognition of writing activity in continuous stream of motion data. *IEEE Transactions on Human-Machine Systems*, 46(3):436–444, 2016.

[2] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324, 2007.

[3] S. S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, 2015.

[4] G. A. ten Holt, M. J. Reinders, and E. Hendriks. Multi-dimensional dynamic time warping for gesture recognition. In *Thirteenth annual conference of the Advanced School for Computing and Imaging*, vol. 300, p. 1, 2007.