TACTILE GESTURE RECOGNITION FOR PEOPLE WITH DISABILITIES

Yu Yuan, Ying Liu, Kenneth Barner

Department of Electrical and Computer Engineering,
University of Delaware, Newark, DE 19716
{yyuan, liuy, barner}@udel.edu

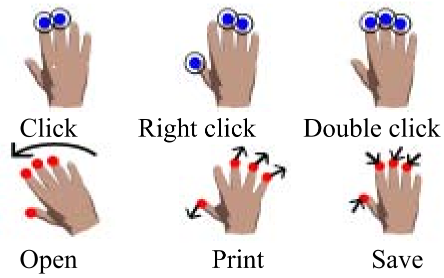
ABSTRACT

Multi-Touch technology provides a successful gesture based Human Computer Interface. The contact and gesture recognition algorithms of this interface are based on full hand function and, therefore, are not accessible to many people with physical disability. In this paper, we design a set of command-like gestures for users with limited range and function in their digits and wrist. Trajectory and angle features are extracted from these gestures and passed to a recurrent neural network for recognition. Experiments are performed to test the feasibility of gesture recognition system and determine the effect of network topology on the gesture recognition rate. These results show that the proposed method can successfully recognize those designed gestures for disabilities.

1. INTRODUCTION

Hand gestures were one of the first and most powerful means of communication, which is established long before speech and language developed. Accordingly, many new Human Computer Interfaces attempt to use gestures as input to communicate and interact with computers. Most recently, multi-touch technology [1] has been successfully introduced. The heart of this technology is a unique, 2-dimensional parallelogram electrode array that produces proximity images of fingers and hands touching on the surface. Software recognizes and tracks all finger motion. Hand gestures are designed by finger combination and a simple motion direction at the gesture starts. The multi-touch pad for gesture and some examples of finger gestures are shown in Fig. 1.

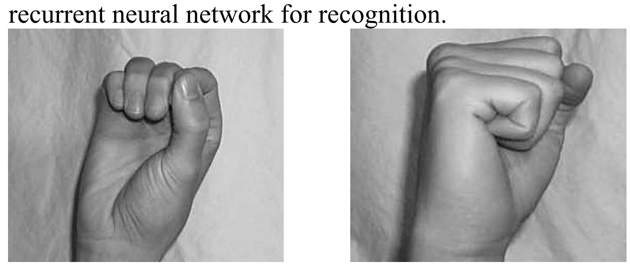
(a)



(b)

Figure. 1 (a) The multi-touch pad for gesture. (b) Examples for the defined gestures

Current finger gestures are well designed for users with full finger function. However, there are many users with physical disabilities who cannot successfully, or repeatedly, perform these gestures. Such limiting conditions include Cerebral Palsy, Muscular Dystrophy and spinal injuries or disorders. Also the elderly population may have symptoms of Parkinson's Disease, stroke and arthritis, which may limit their finger function. The possible hand contact images of the people under above conditions are in Figure 2. Figure 2 (a) shows a person with no finger function and finger contractures, which may be caused by paralysis. Figure (b) gives the contact image of a person with limited finger and wrist function allowing only limited contact with the surface. For these people, they can only perform gestures with partial palm(Fig.2 (a)) or pinky side of the hand(Fig.2 (b)). Motivated by these special requirements, we designed a set of gestures that can be performed using partial palm and the pinky side of the hand. We extract features from the contact images, consisting of hand trajectory and gesture angles. These are then feed into a recurrent neural network for recognition.



(a) (b)

Fig.2 (a) Finger contractures (partial palm). (b) Limited finger and wrist function and range (pinky side).

The remainder of this paper is organized as follows. Section 2 includes two parts: the first part gives the defined hand gestures and feature extraction method; the second part presents the recurrent neural network for gesture recognition. The experiments' results are shown in section 3. Finally, section 4 describes the summary of this paper and high lights future work to be done.

2. HAND GESTURE RECOGNITION

2.1. Gesture Design and Feature Extraction

Some gesture recognition systems [2] use gesture action commands consisting of a subset of American Sign Language (ASL) gestures. A limitation of this approach is that users must start in the designated start position upon initialization of the system. The current gestures designed for the Multi-Touch pad are based on the combination actions of different fingers. This causes difficulty for some users. Thus the challenge addressed here is the design of gestures that (1) can be easily operated by users with physical disability and (2) have distinct 2D features they can be recognized by the neural network system. Under these considerations, we chose command-like and single-stroke gestures as our example vocabulary. These gestures can be performed anywhere on the multi-touch pad.

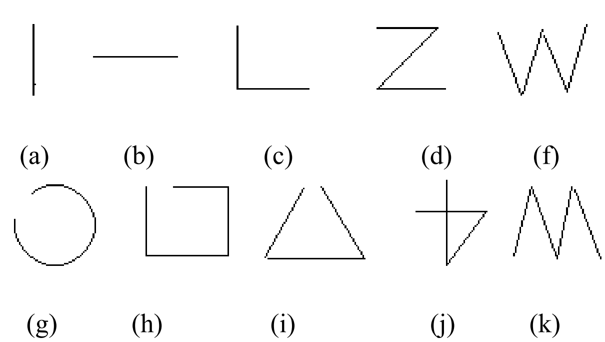


Fig. 3 Examples of designed gesture

There are two main reasons to use single stroke gesture. First, single stroke gestures are more easily performed using partial palm and pinky side, and they coincides with a physical process of tensing and releasing the hand. Second, this choice does not require the segmenting of multi-stroke gesture and shorter timeouts to be used. These gestures are also command based. For example, (a) in Fig. 3 can be interpreted as "up-scroll" and (b) as "forward" when browsing the web page. Similarly, (d) represents deleting a file and (f) opening a file. These command-like gestures can also be extended to 3D [3].

As a gesture is performed on the multi-touch pad, the proximity image is recorded by a set of sensors, as shown in Fig. 3. The pad contains a 16×40 parallelogram electrode array. This array scans all electrodes every 20 ms, producing 50-frame per second (fps) stream of proximity images. A set of features is extracted from the images that are used as input information to the neural network. Many features are defined in [4] to identify gestures. In the case considered here, to balance performance and computational load, we chose the trajectory and angle information as the features.

(a) (b)

Fig. 4 16×40 proximity image of the user's hand recorded from the multi-touch pad (a) Pinky side of the hand. (b) Partial palm.

We apply the moment algorithm to calculate the features of the hand gestures. The computation of image moments, which involves suming over all pixels in the image, provides useful summaries of global image information. Thus this algorithm is robust against the small changes of pixel value [5]. The centroid of the touching hand depends on the zeroth and first moments of the image. The equations define the necessary image moments, where $I(x, y)$ is the image intensity at position(x, y).

$$
\begin{aligned}
M &= \sum_x \sum_y I(x, y) & M\_xy &= \sum_x \sum_y xyI(x, y) \\
M\_x &= \sum_x \sum_y xI(x, y) & M\_y &= \sum_x \sum_y yI(x, y) \\
M\_x^2 &= \sum_x \sum_y x^2 I(x, y) & M\_y^2 &= \sum_x \sum_y y^2 I(x, y)
\end{aligned} \tag{1}
$$

The centroid position G_x, G_y and θ can be determined from the above moments,

$$
G_x = \frac{M\_x}{M} \qquad G_y = \frac{M\_y}{M} \tag{2}
$$

Also of importance is how the hand lies in the field of view i.e, it is initial orientation. This orientation can be defined by the direction of the axis of least inertia. In order to obtain the minimized inertia, we define the intermediate variables *a, b, c* (c.f. [5])

$$a = \frac{M\_x^2}{M} - G_x^2 \qquad b = 2(\frac{M\_xy}{M} - G_x G_y)$$
$$c = \frac{M\_y^2}{M} - G_y^2 \tag{3}$$

The initial orientation of the hand gesture is

$$\theta = \frac{\arctan(b, (a-c))}{2} \tag{4}$$

We also consider the angles between points on the trajectory:

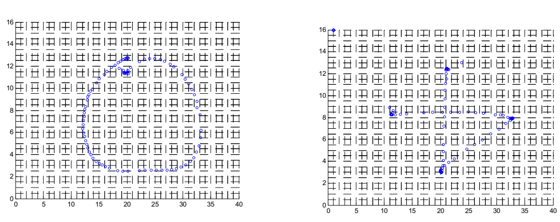
$$\theta_p = \arctan\frac{\Delta x_p \Delta y_{p-1} - \Delta x_{p-1}\Delta y_p}{\Delta x_p \Delta x_{p-1} + \Delta y_p \Delta y_{p-1}} \tag{5}$$

where p is the index of the different points.

The trajectories for a given tactile gesture are analyzed and gesture specific information that describes its behavior is calculated. The descriptive features used are (1) the cosine (f_1) and sine (f_2) of the angles of the gestures, which avoid a discontinuity as the angle passes through 2π and wraps to 0 [2], (2) the sum of the absolute value of the angle at each selected point (f_3), and (3) the total angle traversed (f_4). These extracted features are independent of the overall image intensity.

$$f_1 = \cos(\theta_p) \qquad f_2 = \sin(\theta_p)$$
$$f_3 = \sum_{p=1}^{N-2}\theta_p \qquad f_4 = \sum_{p=1}^{N-2}|\theta_p| \quad p = 1, 2, ..., N \tag{6}$$

Figure 5 shows the trajectory extracted from the example gesture (*f*) and (*i*).



(a) (b)

Fig. 5 Trajectory of example gestures (a) The extracted trajectory of gesture (f). (b) The extracted trajectory of gesture(i).

2.2. Recurrent Neural Network for Gesture Recognition

The tactile gesture recognition problem is stated as follows: There is a set of *N* gesture classes, numbered 0 through *N*-1. Each class is specified by example gestures. Given an input gesture *g*, the goal is to determine the class to which *g* belongs.

Input for the recognition process is the motion vector obtained every 100 milliseconds from the multi-touchpad input device that records the tactile gestures, which is represented as a set of position samples within the two dimensional space. Low-pass filter is used to reduce noise and spikes and convert the input to a single-stroke gesture path [6]:

$$y_{i+1} = kx_i + (k-1)y_i, k > 0.5 \tag{7}$$

where y_i and x_i are the filtered and measured values respectively.

Neural networks and related techniques have been applied to gesture recognition tasks with success. A recurrent network is used in preference to a feed-forward network here, since the ordering of the input is important. Specifically, an Elman Network [7, 8] is employed, which is a recurrent network that is robust to variations in the speed of the gestures.

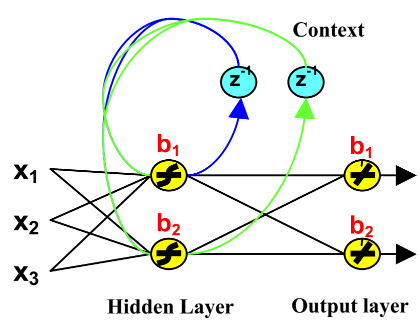


Fig. 6 Structure of the Elman Network (EN)

The network used consisted of 32 input nodes, 20 hidden context nodes, and a 10 output nodes – each output node represents a class of gestures. The inputs to the hidden layer are the present inputs and previously saved outputs of the hidden layer context units. The definition of the network size (the neurons in each layer) is a compromise between generalization and convergence. The processing nodes all used the log-sigmoid logistic function.

The entire input sequence is presented to the network, and its outputs are calculated with the target sequence to generate an error sequence. For each time step, the error is backpropagated to find gradients of errors for each weight and bias. This gradient is, actually, an approximation since the contributions or weights and biases to errors via the delayed recurrent connection are ignored. This gradient is then used to update the weights. The network was trained using the backpropagation through time (BPTT) algorithm [9] with momentum to increase the speed of convergence.

3. EXPERIMENTAL RESULTS

In the recognition phase, we chose a set of 10 common tactile gestures, such as "down", "rectangle", "triangle", "circle", and "zigzag" to determine the recognition performances of the system. For each gesture we use 100 samples (25 for the training set, 75 for testing). The log-sigmoid logistic function is used as the activation function and outputs softmax normalization is performed for classification purpose.

The tactile gesture recognizer described in this paper is implemented in Visual C++, and was subjected to several test inputs. The obtained results are summarized in Table 1 for people with different disabilities.

Table 1 Recognition rates for people with different disabilities

Gesture	Contact image	
	Partial Palm	Pinky Side
Up	100%	100%
Right	100%	100%
Down right	88%	96%
Zigzag	98%	100%
"W"	94%	96%
Circle	87%	90%
Rectangle	93%	96%
Triangle	88%	91%
Down left cross	94%	96%
"M"	88%	96%
Average	93.0%	96.1%

As can be seen from these results, the proposal approach that we adopted fields promising recognition rates, with an overall correct rate of approximately 94.5%.

4. CONCLUSION AND FURTHER WORK

In this paper, we have proposed a tactile gesture recognition system using a recurrent neural network approach. Experiments were carried out on test subjects with two different disabilities, fielding recognition rate of 93.0% and 96.1%. We are currently extending this method to determine what part of the hand is touching the surface of a multi-touch pad. We expect this extension to contribute to even further improvement in recognition performance. Another future direction of this work is an extension to 3D tracking. Currently, our tracking method is limited to 2D motion. While this is enough for our multi-touch pad interface system, interaction based on 3D motion of hands and fingers is necessary for other types of applications.

5. REFERENCES

[1] Westerman, W. and Elias, J., "Multi-touch: A new tactile 2-d gesture interface for human-computer interaction", Proceeding of the Human Factors and Ergonomics Society 45th Annual Meeting, pp. 632-636, 2001.

[2] J. Davis, M. Shah, "Visual gesture recognition", IEE Proc. Vis. Image Signal Process, Vol.141, No.2, pp. 101-106, April 1994.

[3] Nam, Y. and Wohn, K., "Recognition of space-time hand gestures using hidden Markov model", ACM Symposium on Virtual Reality Software and Technology, 1996

[4] Dean Rubine, "Specifying gstures by eample", Computer Graphics, Vol.25, No. 4, pp. 329 – 337, Jul. 1991.

[5] B.K.P. Horn. "Robot vision", MIT Press, 1986.

[6] David J. Sturman, "whole hand input", PhD thesis, MIT, February, 1992

[7] Gurney, K., *An Introduction to Neural Networks*, UCL Press, University of Sheffield, 1997.

[8] Morgan, D. and Scolfield, C., *Neural Networks*, Palgrave, New York, 2000.

[9] Verbos, P.J., Backpropagation Through Time: What it does and how to do it, *Proceeding of the IEEE*, vol. 78(10), pp. 1550-1560, 1990