Jurnal Teknologi

Full paper

Writing in the Air Using Kinect and Growing Neural Gas Network

Mohammad Reza Aminian Heidari*, Azrulhizam Shapi'i, Riza Sulaiman

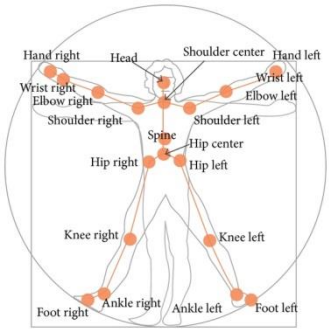
Universiti Kebangsaan Malaysia, Bangi, Malaysia

*Corresponding author: aminiyan.it@gmail.com

Article history

Received : 15 August 2014
Received in revised form :
15 October 2014
Accepted :15 November 2014

Graphical abstract



Abstract

This paper discusses an approach which helps us to recognize English language characters which are written in the air by hands. This method is done by using Kinect camera and growing neural gas network. The proposed character recognition method has three main steps: preprocessing, training and recognition. The system and the proposed method can be considered from two aspects: (a) runtime, and (b) accuracy. One of the main goals in this method is to provide noise tolerance which is necessary for these kinds of methods. IN addition, it has influence upon accuracy rate because the proposed method can remove more outliers. The results show that the proposed method provides good results with the accuracy rate of 95.54%, 97.86% and 99.08% for lower case letters, upper case letters and digits respectively.

Keywords: Kinect; growing neural gas; multi-layer perceptron network

■1.0 INTRODUCTION

Since 3D sensors were introduced to the world of electronics and computers, so many applications were made in monitoring, mapping and navigation areas. However, their high costs prevented them from finding a way to wide spread application and global markets.

When the Kinect sensor was invented by Microsoft, it made a big evolution in the area of 3D based applications. We can use Kinect camera in character recognition research area. In our proposed approach we are going to present a method for character recognition based on Growing Neural Gas and multi-layer perceptron networks which are types of Artificial Neural Networks.

One of the most classical applications of the Artificial Neural Network is the Character Recognition System. This system is the base for many different types of applications in different fields, many of which we use in our daily lives. Cost effective and less time consuming, most businesses, post offices, banks, security systems, and even the field of robotics could employ this system as the base of their operations. Handwritten character recognition is a difficult problem due to the great variations of writing styles, different size (length and height) and orientation angle of the characters. Handwritten character recognition is an area of pattern recognition that has become the subject of research during the last some decades[1]. Character recognition has always been given considerable attention. Because of the fast pace of development in automation, many old methods have been improved to go with the new techniques for recognizing Latin and Chinese characters[2, 3].

Neural network is playing an important role in handwritten character recognition. Many reports of character recognition in English have been published but still high recognition accuracy and minimum training time of handwritten English characters using neural network is an open problem. Therefore, it is a great important to develop an automatic handwritten character recognition system for the English language[1]. On the other hand, online handwriting character recognition is one of the most difficult and complex research problems[4, 5, 6] because of different size and styles of the characters which are written by different people. In this paper, efforts have been made to develop a handwritten character recognition method for English language with high recognition accuracy and minimum runtime.

■2.0 THE PROPOSED METHOD

2.1 Creating the Dataset

In this research, a dataset is needed for testing, validating and training sections. Since the user is writing in the air, the current OCR datasets are not useful for this research because there are some hand written samples which are written on a paper or any other surface.

For creating the dataset, 100 people were asked to write the English language characters by the use of the software which was made in this research. The participants were supposed to write the characters like they are writing them in the air. When you are writing on a paper or any other surfaces you can pick up your hand whenever it is necessary and continue writing but when you are writing in the air you cannot put your fingers up and the camera cannot recognize it. Figure 1 shows a sample of character A which is written by a user.

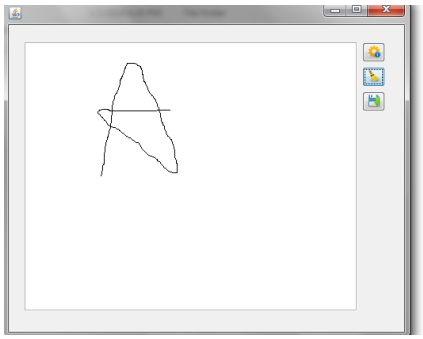


Figure 1 A sample of hand writing

2.2 Recognition and Tracking the Hand

We use Kinect for Windows SDK[7] as a development environment of Kinect sensor. By the use of depth data it is possible to detect and recognize the joint position of the body. Kinect sensor can track 20 places of joint position and can easily carry out hand detected and gesture recognition from this dataset. Figure 2 illustrates the joint that can be detected by Kinect.

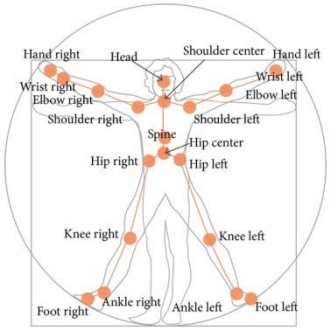


Figure 2 Recognition of the human skeleton (from Microsoft Kinect for Windows SDK).

2.3 Normalization

The data which are obtained in the tracking part are so noisy. These noises are categorized in 3 following categories.

1. The hand tracking algorithm of Microsoft Company is not optimal. It means there are some mistakes and misdetection in tracking the hand movements and these mistakes are illustrated as some noises and extra lines on the screen. So these errors make the path obtained in the hand tracking algorithm noisier.
2. In this system the user draw the characters in the air, so there is high probability that user have extra and unwanted movements. So the path may have extra curves and lines. These extra curves and lines are also considered as a noise.
3. Hands movements should start from one point and ends to another (end point) point. Therefore, all the movements before the start point and after the end point are considered as useless data.

Because of the Noises and outlier data which mentioned above, there is a need to eliminate such kind of data from the obtained path. Eliminating the useless data has two steps: (1) constructing the region of interest; and (2) Removing noisy data.

2.4 Constructing Region of Interest

During the extracting hand movements in this system, a boundary is being determined so that the user can draw the character within this boundary. This boundary is a Tetrahedral with predefined measures and the user can change the size of the edges of this tetrahedral in the settings option. All the curves and lines out of this boundary are being ignored. Following is the Pseudocode for region of interest in Figure 3.

```
REGION OF INTEREST METHOD (arr, len),
Min_x <- arr_x(1)
Min_y <- arr_y(1)
Max_x <- arr_x(1)
Max_y <- arr_y(1)

For each point in sampling points (index <- 1 ~ len)
   If arr_x(index) is lower than min_x then
      min_x <- arr_x(index)
   If arr_y(index) is lower than min_y then
      min_y <- arr_y(index)
   Else If arr_x(index) is higher than max_x then
      max_x <- arr_x(index)
      Else If arr_y(index) is lower than max_x then
         max_x <- arr_y(index)
End

For each point in sampling points (index <- 1 ~ len)
   if arr(index) is not in defined criteria then
      remove arr(index)
End

END REGION OF INTEREST METHOD
```

Figure 3 Region of interest pseudocode

2.5 Noise Reduction

For reducing the noisy data using this approach, a normalization technique with first derivation degree is used. First the first derivation of considered curve is being calculated which gives back the slope between the sampling points directly. The result is the amount of change in the direction of the curve in each moment. These results can help the system to delete the sampling points which have high amount of change in their location.

To omit the environment noises, the first derivation of the extracted path is being calculated to achieve the slope between sampling points. Bye use of the first deviation results, system computes the angle of changes.

In this technique, two thresholds are chosen then the calculated angles are being normalized and sit through 0 to 360. After that, all the sampling points which are located in defined thresholds will be eliminated. By this normalization process, all

the small environmental noises and breaks are eliminated. Figure 4 shows the Pseudocode of noise reduction step.

```
REDUCE NOISY SAMPLING POINTS (arr, len)

arr <- array of sampling points
len <- SIZE(arr)

arr_m (1) <- 0
for each sampling point in the array (index <- 2 ~ len),
        point2 <- arr (index)
        point1 <- arr (index - 1)
        m = (point2(y) - point1(y)) / (point2(x) - point1(x))
        arr_m (index) <- m
end
for each sampling point in the array (index <- 2 ~ len)
        m2 <- arr_m (index)
        m1 <- arr_m (index - 1)
        angle <- arctan((m1 – m2)/1+m1m2)
        if angle is inside the predefined criteria,
                delete index th point in the array
        end
end

END REMOVE NOISY SAMPLING POINTS METHOD
```

Figure 4 Remove noisy sampling points pseudocode

In this technique, an interval will be defined by the user. If the angle between two lines be in the defined criteria, the sampling point, which connects these two lines, will be eliminated and the pervious point will be connected to the current point.

In this normalization technique, the value of first deviation of extracted path which contains so many sampling points is calculated. So after calculating the first derivation, the slope between each two lines is available. In continue, the slopes will be investigated and the local angel for each sample point will be calculated. By the following formula we are going to calculate the slope between two points.

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\theta = \arctan\left(\left|\frac{m_1 - m_2}{1 + m_1 m_2}\right|\right), \quad 0 \le m_{angle} \le \frac{\pi}{2}$$

$$m_1 m_2 = -1 \; when \; the \; lines \; are \; perpendicular \; so \; \theta = \frac{\pi}{2}$$

In this technique, each achieved angel comparison with the pervious angel. If the difference between them is lower than threshold which is defined as 26 in this system, the calculated point will be eliminated and the pervious point will connect to the next point.

There are two steps after eliminating the noisy sample points: (1) extracting the key-features; and (2) choosing limited numbers of key-features. Specifying the key-features helps the character recognition algorithm to detect the considered character and ignore processing the useless data. In this research to find these key-features, a down-sampling algorithm based on growing neural gas technique is used.

The achieved sampling points from the noise reduction part are transferred as an input to down-sampling algorithm. Then the growing neural gas algorithm starts to add the key-features to the points and change their current position to the right position. According to the congestion of the sampling points in different locations, the growing neural gas algorithm defines and locates the key-features.

2.6 Growing Neural Gas

Growing neural gas is an unsupervised learning method. It means that there is no information about the output[8]. The growing neural gas also is able to recognize the number of neurons needed to describe an input dataset. In this algorithm, at first there are only two neurons in the system, then the input data are being feed to growing neural gas.

According to the location of input data, the location of neurons will be changed to reduce the errors of the system. After having specified number of input signals, a neuron will be added to system in order to reducing the average local errors of the system.

The above process is being executed repeatedly until the number of neurons reaches the predefined number of them. The black points are curve's sampling points and the red one is GNG neuron's centroid. Figure 5 shows a neuron and its neighbors and Table 1 shows the parameters and their value of GNG algorithm.

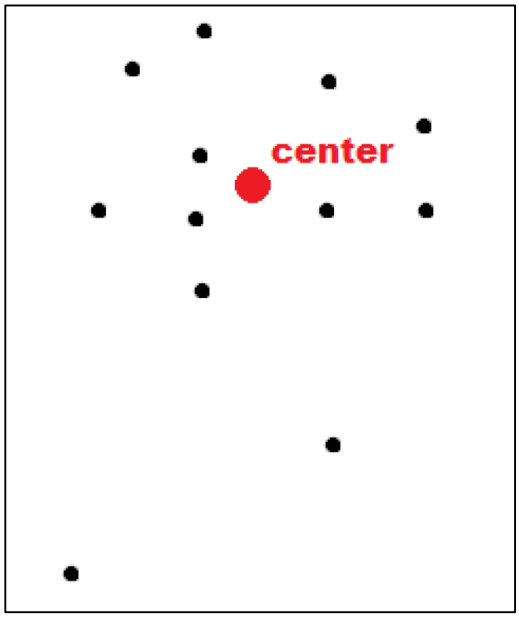


Figure 5 A neuron and its neighbors

Table 1 Defining the parameters for neural gas algorithm

Learning rate for winner's neighbors	0.05
Learning rate for winner	0.0006
Threshold for edge removal	21
Number of signals for each neuron insertion	32
Local error's increasing rate for two neurons having highest local error rate	0.5
Local error's increasing rate for rest of neurons	0.00049
Final number of neurons	100

The value of the aforementioned parameters is chosen based on trial and error method. It means in different experiments they had the best outcome.

The most important parameter in down-sampling technique is the number of signals which are needed to add one neuron to growing neural gas in the process. The quality of down-sampling results is related to the mentioned parameter. Also it has a

considerable effect on the down-sampling running time. The execution time and the quality of results in down-sampling part are related to the parameters which are defined in growing neural gas network.

2.7 Character Recognition

After normalization part and gathering the key-features, now all the data to execute the character recognition part are available. In character recognition part, a multi-layer perceptron (MLP) neural network is used. An MLP consist of at least 3 layers, an input layer, one or more hidden layer(s) and an output layer. The input layer is not considered a "true" layer because no computation is performed by it. It receives problem-specific inputs from the outside world. An MLP contains one or more hidden layers, which receive inputs from preceding layers (input or hidden layers) and their outputs connect to the next layers (output or hidden layers)[9]. Each neuron in a hidden layer employs a nonlinear activation function that is differentiable[10]. Initially in the setup part, by the use of the defined dataset, the neural network is being trained to recognize the characters. There neural network receives the key-features as an input and defines the percentage of dependency of this set to each character. The character which has the highest dependency among other characters is chosen as the best candidate. Table 2 illustrates the MLP features and their value.

Table 2 MLP features

Input layer	100 neuron
Hidden layer	5151 neuron
Output layer	26 neuron (for each character)

3.0 RESULTS AND DISCUSSION

3.1 Runtime Evaluation

Table 3 shows the running time for some samples of the dataset.

Table 3 Runtime evaluation

	Number of sampling points	Runtime
1	700	130 msec
2	1012	154 msec
3	450	50 msec
4	2100	200 msec
5	500	110 msec
6	605	110 msec
7	443	49 msec
8	1500	180 msec
9	550	111 msec
10	716	130 msec
11	802	143 msec
12	1002	162 msec
13	1402	178 msec
14	1301	170 msec
15	1512	181 msec
16	805	143 msec
17	901	143 msec
18	1000	162 msec
19	2019	203 msec
20	301	40 msec

As shown in Table 3 the runtime execution has encouraging results.

3.2 Accuracy Rate

The proposed method has been tested for upper case, lower case and digits. Table 4 shows the results.

Table 4 Accuracy rate

lower case	95.54%
upper case	97.86%
Digits	99.08%

4.0 CONCLUSION

In this paper, a method was presented for recognizing the characters which are written in the air by hands. Kinect camera recognizes and tracks the hands movements. The achieved path will be fed into the proposed OCR engine which contains growing neural gas and multi-layer perceptron networks. A dataset gathered from 100 people in order to test, validate and train the system was used in this study.

The experiment results show that the proposed approach is applicable for recognizing the particular movements which are captured by Kinect camera. In the future, the proposed method can be used in other OCR techniques to increase the accuracy rate and decrease the running time.

References

[1] V. Patil, S. Shimpi. 2011. Handwritten English Character Recognition Using Neural Network. *Elixir Comp. Sci. & Engg*. 41: 5587–5591.

[2] M. Fahmy, H. El-Messiry. 1999. Zernike Moments As Feature Extracto for Arabic Character. The 1st-MINIA International Conference for Advanced Trends in Engineering, March 14–17.

[3] D. Shi, S. R. Gunn, R. I Damper. 2003. Handwritten Chinese Radical Recognition Using Nonlinear Active Shape Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 25(2): 277–280.

[4] R. Plamondon and C. M Privitera. 1999. The Segmentation of Cursive.

[5] Handwriting: An Approach Based on Off-Line Recovery of the Motor-Temporal Information. *IEEE Trans. Image Processing.* 8(1): 80–91.

[6] R. Plamondon , and Sargur N. Srihari. 2000. On-Line and Off-Line.

[7] Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions*

[8] *on Pattern Analysis and Machine Intelligence.* 22(1): 63–84.

[9] L. Xiaolin, and D.-Y Yeung,. 1997. On-line Handwritten Alphanumeric Character Recognition Using Dominant Points In Strokes. *Pattern Recognition.* 30(1): 31–44.

[10] Kinect for Windows SDK 2.0 Public Preview: http://www.microsoft.com/en-us/download/details.aspx?id=43661.

[11] G. Tesauro, D. S. Touretzky and T. K. Leen 1995. *Advances in Neural Information Processing Systems 7.* MIT Press, Cambridge MA.

[12] G. Zhenning 2013. An Abstract of Parallel and Distributed Implementation of A Multilayer Perceptron Neural Network on A Wireless Sensor Network, Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Engineering, The University of Toledo.

[13] O. Fink, E. Zio, & U. Weidmann, 2013. Predicting Time Series of Railway Speed Restrictions With Time-Dependent Machine Learning Techniques. *Expert Systems with Applications*. 40(15): 6033–6040.