A Novel Recognition System for Digits Writing in the Air using Coordinated Path Ordering

Chiang Wang
Dept. of Electrical Engineering
National Taiwan Normal University
Taipei, Taiwan
40175033H@ntnu.edu.tw

Chung-Yen Su
Dept. of Electrical Engineering
National Taiwan Normal University
Taipei, Taiwan
scy@ntnu.edu.tw

Chun-Lin Lin
Dept. of Electrical Engineering
National Taiwan Normal University
Taipei, Taiwan
60375034H@ntnu.edu.tw

***Abstract*—With the invention of Microsoft Kinect sensor, human-computer interaction is gaining its attention and becoming available for widespread use. The previous study presented a method of Kinect-based mid-air handwritten digit recognition for Android smart phones with a recognition accuracy of only about 94.6%. In this paper, we propose an improved method based on the normalizing and scaling of path order coordinates. With that, the proposed method leads to accuracy elevation and executing time reduction. Experimental results show an average recognition accuracy rate of 96.8% was achieved for each number.**

Keywords—Kinect sensor; human-computer interaction; handwritten digit recognition; accuracy rate; support vector machine

I. INTRODUCTION

Human-computer interaction is a prevailing research topic. The development of control-free applications such as in the field of gaming [1], education or even health care [2] is growing rapidly. People can communicate with or declare instructions toward machines by simply hand gesturing [3] or posing, providing users a much more intuitive and novel way. The traditional human-computer interaction is based on optical cameras, which are unable to provide depth information on an object. This leads to limited applications. However, with the alternative approach of Microsoft Kinect, an infrared light range-sensing camera, developers can easily gain depth information. By using depth extraction technic, some researchers modeled fingertips movements and hand gesture [4, 5].

Kinect (see Fig. 1) entered the market in 2010, which consists of an RGB camera and a depth sensor. In 2011, Microsoft Software development kit (SDK) is released, providing developers to create Kinect applications. For example, an effective algorithm of handwritten digit recognition for handwritten digit recognition using multiple segments and scaled coding for dialing numbers and setting a timer was developed [6]. However, the recognition accuracy of the system was only approximately 94.6%. Therefore in this study an improved algorithm will be introduced, increasing the recognition accuracy up to 96.8%.

Fig. 1. Kinect: the RGB and Depth sensor.

Handwriting recognition has become a compelling research area in field of image processing and human-computer interaction in the recent years. Several research works have been focusing on new methods with reduction of processing time while simultaneously providing higher recognition accuracy.

Recognition of handwritten digits written in the air is more challenging comparing to digit written on paper due to the latter doesn't have to take motion path order into account. In Kinect-based recognition, strokes might overlay. This causes crucial disturbance to the recognition of digits. However, the former have been shown to be superior in recognizing handwritten digits due to the temporal information on the former. In Kinect-based recognition system, the order of strokes made by the user are available.

Qu, Zhang and Tian [7] proposed an online Kinect handwritten digit recognition using fingertip tracking, offering a high recognition accuracy and robustness to noisy data test with small training number, but with complex feature extraction including movement velocity values and path angle.

In the previous study, Kinect-based mid-air handwritten digit recognition using multiple segments and scaled coding is proposed [8], however recognition rate could be further improved.

In this paper, a Kinect-based recognition system for digits written in the air using coordinated path order with normalization coding is proposed. The original pre-processed coordinates are generated by using the Microsoft Kinect SDK to track user's body skeleton, recording the motion path of upper body joints, both hands included, for further analysis. Each set of coordinate is normalized into a 6 × 8 table and these feature points are used to train libsvm [9]. Experimental results show that the proposed recognition system provides higher recognition accuracy.

This paper is organized as follows. In section II, the

978-1-4799-8562-3/15/$31.00 ©2015 IEEE

Kinect-based digit recognition system is presented. Section III presents the proposed coordinated path order with normalization coding. Next, Section IV presents the experimental results and comparative analysis and finally, the paper is concluded in section V.

II. The Proposed Recognition System

The flowchart of the proposed recognition system for digits written in the air is illustrated in Fig. 2. First, Kinect is initialized, including turning on the image stream and depth stream. Kinect SDK is able to offer developer the coordinates of joint position(X, Y, Z), and we can use skeleton tracking engine to get joint information. Then we retrieve specific joints to judge if the "start" condition is satisfied.

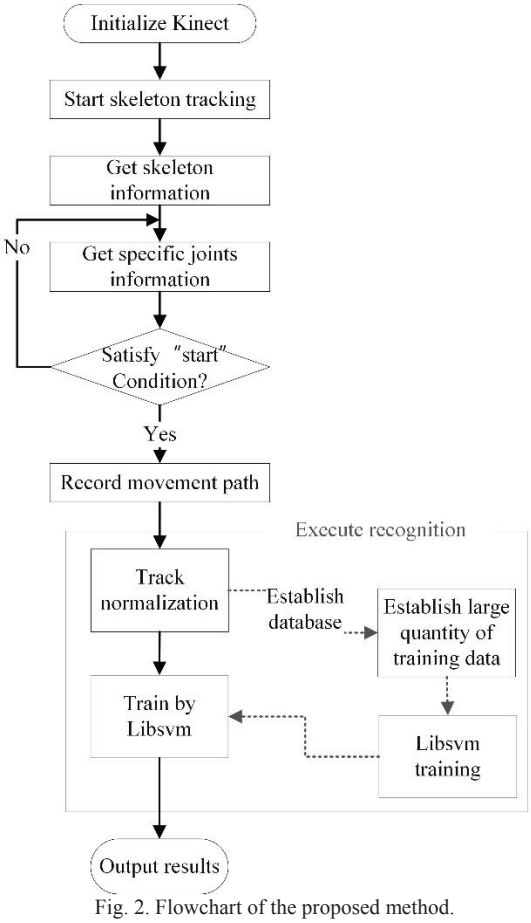


Fig. 2. Flowchart of the proposed method.

Microsoft Kinect SDK can detect human skeletal information 30 frames per second in real time. Here, we set the recording time to be three seconds, which result in 90 coordinates. Distancing the Kinect sensor form 850mm to 4000mm, coordinates of strokes are recorded into a 640 pixels × 480 pixels window, the coordinates are then normalized into a 6 × 8 table. Each coordinate corresponds to a certain normalized coordinate in the 6 × 8 table. According to the stroke sequence, points are encoded to generate the normalized (x, y) coordinate set.

Finally, we use libsvm, developed by Prof. Chih-Jen Lin, for the recognition of digits. The input format for libsvm is described as below.

[label] [index1]:[value1] [index2]:[value2]…
[label] [index1]:[value1] [index2]:[value2]…
…

Label signifies the class in our classification; it is the digit of "0" to "9". Value is the data for training or predicting. The index here indicates the feature points while each normalized x and y coordinate are placed into value$i$ and value(i+1) sequentially. After generating a model after the off-line training, we can then perform the recognition based on the model and obtain the result in real-time.

In this paper, we focus on the improvement of algorithm, which generates different sets of features for each digit for both the training and testing of our dataset.

The most difficult part for the recognition of handwritten digits in the air is the overlapping of strokes, and the disturbance of superfluous strokes, such as digit "1", "4" and "5" shown below. Those will all be inevitably classified as feature points.

The previous method concatenates the shape code and the path code to form the features of a digit. This method is able to compensate flaws between the shape and path code, and has great recognition ability. However, this method may cause confusion, accuracy rate could be further ameliorated. Since some digits, such as "5" and "3", shown in Fig. 3, have similar writing style, that is to say, similar shape code. The remaining 48 features, been segmentally processed, aren't able to give a precise path of the handwritten digit. Recognition is likely to fail.

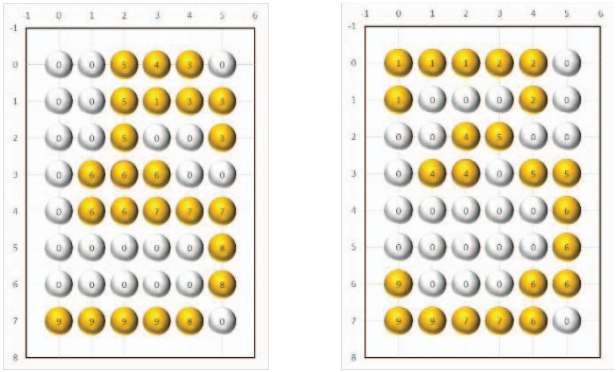
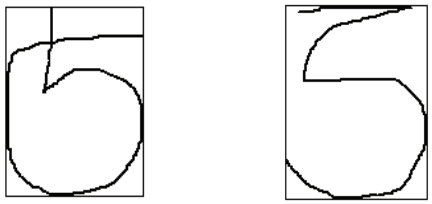


Fig. 3. An example of the digits "5" and "3" with similar shape codes but different path codes.

Observing from the testing dataset, we can generalize a conclusion that digit "5" has two kinds of writing styles, as shown in Fig.4. While Fig.4 (b) is prone to cause confusion comparing to Fig.4 (a), since digit "5" resembles the feature points of digit "3". False judgment is likely to occur between digit "7" and digit "9" too, reason likewise.

978-1-4799-8562-3/15/$31.00 ©2015 IEEE



(a) (b)

Fig. 4(a) and fig. 4(b) presented two different kinds of examples of possible writing styles of digits "5".

III. Coordinated Path Ordering with Normalization Coding

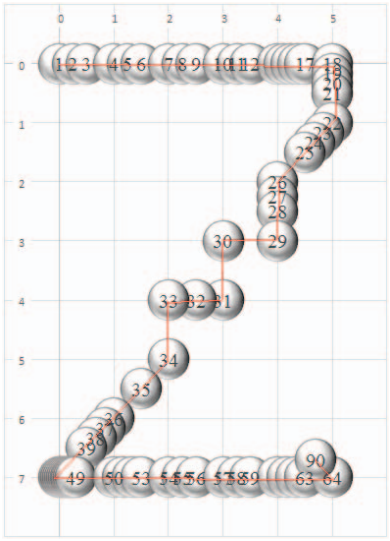


Fig. 5. An example of the proposed coding.

Fig. 5 illustrates our proposed coordinated path order with normalization coding. Features are recorded as a temporal data series in time order. Every point in each kind of coding are combined by a set of two-dimensional coordinate. This algorithm may generate different sets of 180 features which is equal to resulting 90 coordinates for the same digit due to different writing styles. We map the coordinates onto a 6x8 table according to stroke orders and then encode the path code of a digit.

However, the proposed coordinated path order with normalization coding has more feature points than the previous, and the promotion of accuracy rate is expected. We go further in reducing the feature points of each digits into 90, 60, 36, 30, 20, 18, 12, 10 feature points, taking one point out of every two or three feature points and so forth. Then explore the effect of different quantity of feature points to study which results in the best recognition rate.

Example of the proposed method to form features codes of digit "2" are listed as below:

The original path order coding.
(0,0) (0,0) (0,0) (1,0) (1,0) (1,0) (2,0) (2,0) (2,0) (3,0) (3,0) (3,0) (4,0) (4,0) (4,0) (4,0) (4,0) (5,0) … (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7)

Picking out one coordinate point out of every two points makes a 90 feature point coding.
(0,0) (0,0) (1,0) (2,0) (2,0) (3,0) (4,0) (4,0) (4,0) (5,0) (5,0) (5,1) (5,1) (4,2) (4,3) (3,4) (2,4) (2,5) (1,6) (1,6) (0,7) (0,7) (0,7) (0,7) (0,7) (1,7) (1,7) (2,7) (3,7) (3,7) (4,7) (4,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7)

Picking out one coordinate point out of every three points makes a 60 feature point coding.
(0,0) (1,0) (2,0) (3,0) (4,0) (4,0) (5,0) (5,1) (5,1) (4,2) (3,4) (2,5) (1,6) (0,7) (0,7) (0,7) (0,7) (1,7) (2,7) (3,7) (4,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7) (5,7)

Picking out one coordinate point out of every six points makes a 30 feature point coding.
(0,0) (2,0) (4,0) (5,0) (5,1) (3,4) (1,6) (0,7) (0,7) (2,7) (4,7) (5,7) (5,7) (5,7) (5,7)

Picking out one coordinate point out of every nine points makes a 20 feature point coding.
(0,0) (3,0) (5,0) (4,2) (1,6) (0,7) (2,7) (5,7) (5,7) (5,7)

Picking out one coordinate point out of every ten points makes an 18 feature point coding.
(0,0) (3,0) (5,0) (3,4) (0,7) (1,7) (4,7) (5,7) (5,7)

Table I shows the confusion matrix of the previous recognition results of handwritten digits written in the air. We can see that the recognition error occurred between digit "3", "5" and between digit "7" ,"9" both greatly reduced. The improved coordinated path order with normalization coding method gives us a more precise stroke order, the overlapping part will not be overwritten, providing us a more effective way to differentiate digits. This kind of encoding method is able to present both the stroke order and the shape of specific handwritten digit at the same time. Results in a much more improved recognition rate.

At last, we use the libsvm for training and testing. Since the range of the codes of a digit may be too large or too small, the use of scaling parameter in libsvm generates scaled codes, with which will improve the recognition rate.

IV. Experimental Results

The dataset used to train the libsvm consists of 1000 samples (100 sample per digit). After training, we used the already established 2300 testing samples (230 samples per digit) for the testing of our recognition system.

The recognition results are tabulated in Table II, III and IV, while Table III and IV are in the form of confusion matrices, our system achieves a recognition rate up to 96.83%. We can see in these tables, the proposed method has higher recognition than the method in [8] (see Table I).

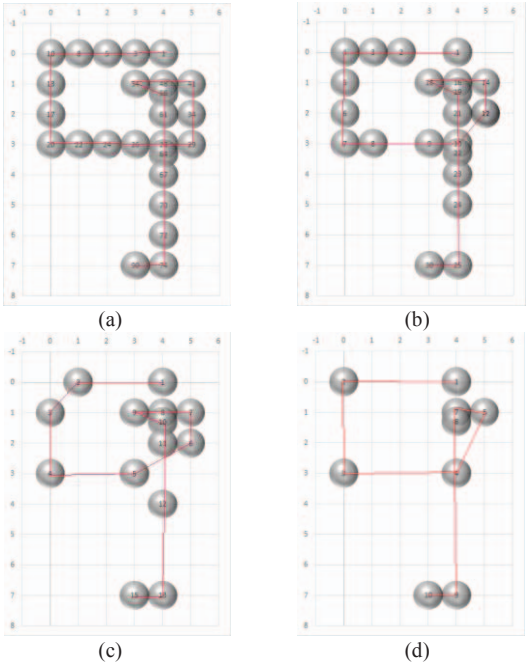
978-1-4799-8562-3/15/$31.00 ©2015 IEEE

TABLE I. ACCURACY OF HANDWRITTEN DIGIT RECOGNITION [8]

input \ output	0	1	2	3	4	5	6	7	8	9
0	225	0	1	0	1	0	2	1	0	0
1	0	218	9	1	0	0	0	1	0	1
2	1	4	220	0	0	2	0	2	0	1
3	0	0	4	215	0	8	0	0	2	1
4	0	0	1	0	221	0	0	1	0	7
5	3	0	0	8	1	205	6	0	7	0
6	3	0	1	1	0	4	220	1	0	0
7	0	5	1	0	2	0	0	217	0	5
8	3	0	2	1	0	8	0	0	216	0
9	0	1	0	1	3	0	0	6	0	219

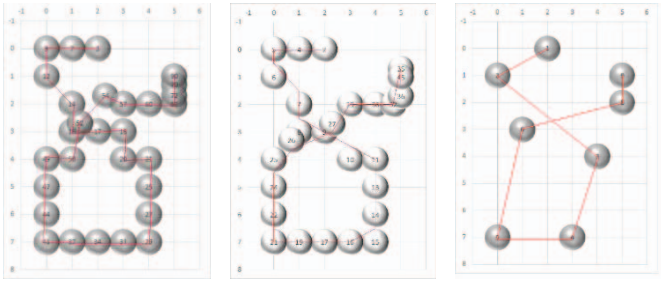
With the cutting down of feature points, the disturbance of the superfluous strokes reduces. Here we present three examples of how the reduction of feature points affect recognition accuracy of digits.

As we can see in Fig. 6(a), the superfluous stroke causes great impact on the recognition of the digit, it results in digit "9" to be misjudged as digit "4", with the reduction of feature points to 60 and under (see Fig. 6(b) to Fig. 6(d)), the influence diminishes. Recognition success. Fig. 7(a) to Fig. 7(c) presents the second example of misjudgment between digit "8" and digit "5". Fig. 7(a) and Fig. 7(b) shows that digit "8" is misjudged as digit "5", this recognition mistake is caused due to the gap between the starting point and the ending point of some dataset of digit "8", this causes the coding resembles digit "5". With the reduction of feature points to 18 (see Fig. 7(c)), the spaces between each point balanced, a correct recognition is brought out. The final example of misjudgment between digit "1" and digit "2" is shown in Fig. 8(a) to Fig. 8(b). When the apogee of digit "1" has more than one feature points, it can easily be mistaken as digit "2", with the reduction of feature points to under 36, the peak point of digit "1" decrease to one point. Another example of recognition success.



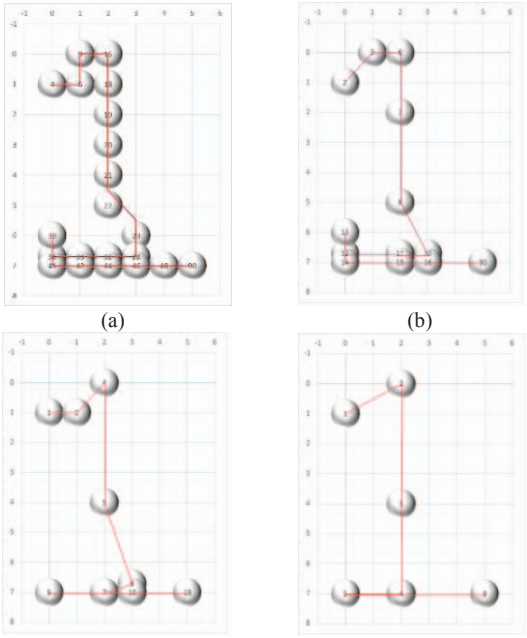
(a) (b) (c) (d)

Fig. 6. An example of the digit "9" recognized as digit "4" (a) false judgement of digit "9" with 180 feature points recognized as digit "4" (b) successful recognition of digit "9" with 60 feature points (c) successful recognition of digit "9" with 30 feature points (d) successful recognition of digit "9" with 20 feature points.



(a) (b) (c)

Fig. 7. An example of the digit "8" recognized as digit "5" (a) false judgement of digit "8" with 180 feature points recognized as digit "5" (b) false judgement of digit "8" with 90 feature points recognized as digit "5" (c) successful recognition of digit "8" with 18 feature points.



(a) (b) (c) (d)

Fig. 8. An example of the digit "1" recognized as digit "2" (a) false judgement of digit "9" with 180 feature points recognized as digit "2" (b) false judgement of digit "1" with 60 feature points recognized as digit "2" (c) successful recognition of digit "1" with 36 feature points (d) successful recognition of digit "1" with 18 feature points.

The accuracy of each digit of different feature points is calculated and shown in Table II. From the table shown, we can sort out that digit "1" is the most difficult digit to recognize, and its accuracy rate drops sharply comparing to the previous method [8]. Observing from the training dataset of digit "1", 10 and more kinds of writing style of digit "1" are classified, while some other dataset from digit "1" are distorted. Without enough training dataset, this leads to a poor recognition result.

978-1-4799-8562-3/15/$31.00 ©2015 IEEE

TABLE II. INDIVIDUAL ACCURACY RATE OF HANDWRITTEN DIGIT RECOGNITION COMPARISON BETWEEN [8] AND PROPOSED METHOD

Digit	0	1	2	3	4
[8]	97.83%	94.78%	95.63%	93.48%	96.09%
prop. (180 feature points)	99.13%	90.87%	96.96%	98.70%	94.78%
prop. (30 feature points)	97.83%	91.30%	96.09%	97.39%	97.39%
Digit	**5**	**6**	**7**	**8**	**9**
[8]	89.13%	95.65%	94.35%	93.91%	95.22%
prop. (180 feature points)	97.83%	96.52%	97.83%	95.65%	98.70%
prop. (30 feature points)	98.26%	97.83%	97.83%	95.22%	99.13%

TABLE III. CONFUSION MATRIX OF HANDWRITTEN DIGIT RECOGNITION OF 180 FEATURE POINTS (PROPOSED METHOD)

input \ output	0	1	2	3	4	5	6	7	8	9
0	228	0	0	0	0	0	0	0	2	0
1	1	209	14	0	0	1	1	2	0	2
2	2	4	223	0	0	0	0	1	0	0
3	1	0	0	227	0	0	1	1	0	0
4	2	1	0	0	218	0	0	1	0	8
5	0	0	0	2	1	225	0	0	2	0
6	5	2	0	0	0	1	222	0	0	0
7	0	4	0	0	0	1	0	225	0	0
8	4	0	0	0	0	6	0	0	220	0
9	0	0	0	0	2	0	0	1	0	227

TABLE IV. CONFUSION MATRIX OF HANDWRITTEN DIGIT RECOGNITION OF 30 FEATURE POINTS (PROPOSED METHOD)

input \ output	0	1	2	3	4	5	6	7	8	9
0	225	0	0	0	0	0	0	0	5	0
1	0	212	12	0	0	1	1	2	0	2
2	2	7	219	0	0	0	0	2	0	0
3	1	0	0	227	0	0	1	1	0	0
4	1	0	0	0	222	1	0	1	0	5
5	0	0	0	1	1	225	0	0	2	1
6	4	2	0	0	0	1	222	0	1	0
7	0	4	0	0	0	0	0	225	0	1
8	4	0	0	0	0	6	0	0	220	0
9	0	0	0	0	1	0	0	1	0	228

The mean accuracies of different amount of feature points are shown in Fig. 9. As we can see in this figure, while feature points gradually decrease from feature points 180 the recognition rate increases with a steady and flat slope until it reaches feature points 30. Once lower than 30 feature points, accuracy drops. Therefore, we conclude that feature points 30 has the highest recognition rate with the highest efficiency.

Feature points	Huang.	180	90	60	36
Accuracy rate	94.6%	96.65%	96.65%	96.83%	96.74%
Feature points	**30**	**20**	**18**	**12**	**10**
Accuracy rate	96.83%	95.83%	96.3%	94.74%	91.35%

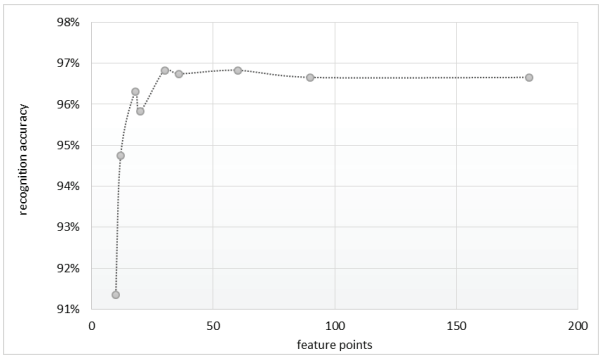


Fig. 9. A scatter chart of mean accuracies of handwritten digit recognition using different feature points from 180 to 10 and comparison with Huang's method [8].

Comparing to the original algorithm, our proposed method has higher accuracy rate and lower executing time.

V. CONCLUSION

A novel coordinated path order with normalization coding and feature extraction method to improve recognition rate of recognition system for digits written in the air has been introduced in this paper for recognizing and classifying digits. We map the coordinates of stroke orders and encode the path code of a digit. Experiment results show the efficiency of the proposed method. It can provide a more efficient method in comparison with the previous [8], with accuracy exaltation and execution time reduction to recognize digits written in the air.

REFERENCES

[1] J. Bobeth, S. Schmehl, E. Kruijff, S. Deutsch, and M. Tscheligi, "Evaluating performance and acceptance of older adults using freehand gestures for TV menu control," Proceedings of the 10th European conference on Interactive tv and video, pp.35-44, 2012.

[2] Bingbing Ni, Nguyen Chi Dat and Moulin, P., "RGBD-camera based get-up event detection for hospital fall prevention," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1405-1408, 2012.

[3] Y. Li, "Hand gesture recognition using Kinect," in the Proc. IEEE 3rd International Conference on Software Engineering and Service Science, pp. 196 - 199, 2012.

[4] L. C. Ebert, G. Hatch, M. J. Thali, and S. Ross, "Invisible touch—Control of a DICOM viewer with finger gestures using the Kinect depth camera," Journal of Forensic Radiology and Imaging, pp. 10- 14, 2013.

[5] Xin Zhang; Zhichao Ye, Lianwen Jin, Ziyong Feng and Shaojie Xu, "A New Writing Experience: Finger Writing in the Air Using a Kinect Sensor," *Multimedia IEEE*, pp. 85-93, 2013.

[6] C. Y. Su, F. A. Huang, C. Y. Shih, and Y. T. Chen, "Kinect-based midair handwritten number recognition system for dialing numbers and setting a timer," in *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pp. 2127-2130, 2014.

[7] Chengzhang Qu, Dengyi Zhang, Jing Tian, "Online Kinect Handwritten Digit Recognition Based on Dynamic Time Warping and Support Vector Machine," *Journal of Information & Computational Science*, Vol. 12, pp.413-422, 2015.

[8] F. A. Huang, C. Y. Su, and T. T. Chu, "Kinect-based mid-air handwritten digit recognition using multiple segments and scaled coding," *International Symposium on Intelligent Signal Processing and Communications Systems*, pp.694-697, 2013.

[9] C. J. Lin. [Online].Available: http://www.csie.ntnu.edu.tw

978-1-4799-8562-3/15/$31.00 ©2015 IEEE