Hand Gesture Recognition Using Kinect via Deterministic Learning

Fenglin Liu, Bangxing Du, Qinghui Wang, Ying Wang, Wei Zeng
School of Mechanical & Electrical Engineering, Longyan University, Longyan 364012, P.R. China
E-mail: {liufenglin45, dubangxing, wangqinghui, wangying, zengwei}@lyun.edu.cn

Abstract: Hand gestures are spatio-temporal patterns which can be characterized by collections of spatio-temporal features. However, in real world scenarios, hand gesture recognition suffers from huge challenges with variations of illumination, poses and occlusions. The Microsoft Kinect device provides an effective way to solve the above issues and extract discriminative features for hand gesture recognition. The recognition approach consists of two stages: a training stage and a recognition stage. In the training stage, hand gesture features representing hand motion dynamics, including spatial position and direction of fingertips, are derived from Kinect. Hand motion dynamics underlying motion patterns of different gestures which represent Arabic numbers (0-9) are locally accurately modeled and approximated by radial basis function (RBF) neural networks. The obtained knowledge of approximated hand motion dynamics is stored in constant RBF networks. In the recognition stage, a bank of dynamical estimators is constructed for all the training patterns. Prior knowledge of hand motion dynamics represented by the constant RBF networks is embedded in the estimators. By comparing the set of estimators with a test gesture pattern to be recognized, a set of recognition errors are generated. The average L_1 norms of the errors are taken as the recognition measure between the dynamics of the training gesture patterns and the dynamics of the test gesture pattern according to the smallest error principle. By using the 2-fold and 10-fold cross-validation styles, the correct recognition rates are reported to be 95.83% and 97.25%, respectively.

Key Words: Hand Gesture Recognition, Deterministic Learning, Kinect, Hand Motion Dynamics, RBF Neural Networks

1 Introduction

Automatic hand gesture recognition has gained increasing attention because of its applications for interactive human-machine interface and virtual environments [1]. Hand gestures can be treated as spatio-temporal actions which are characterized as 3-dimensional (3D) volumetric data. Trajectories are widely used as distinct spatio-temporal features for recognizing not only hand gestures [2], but also action recognition [3]. Lu et al. [4] extracted action patterns via computing position offset of 3D skeletal body joints locally in the temporal extent of video to perform gesture recognition. Zhang and Tian [5] proposed a discriminative 3D descriptor which can effectively capture and model the rich surface shape information of the depth maps. Maqueda et al. [6] designed a hand-gesture recognition system through three stages: detection, tracking and recognition.

Different from the above mentioned approaches, recently, the proliferation of somatosensory camera (such as Kinect [7] and Leap Motion [8]) and smart terminal has dramatically changed the video capture landscape. With the Kinect sensor it can detect and segment the hands robustly, which provides a valid base for gesture recognition [9]. Kinect provides full-body 3D motion capture and directly computes the position of the fingertips. Cheng et al. [10] proposed an Image-to-Class Dynamic Time Warping approach with Kinect for the recognition of both 3D static hand gestures and 3D hand trajectory gestures.

This work was supported by the National Natural Science Foundation of China (Grant no. 61304084), by the Program for New Century Excellent Talents in Fujian Province University, by the Science and Technology Project of Longyan University (Grant nos. LC2015008, LC2015009, LQ2015027), and by the Training Program of Innovation and Entrepreneurship for Undergraduates (Grant no. 2016001).

2 Method

In this section, we propose a method for the recognition of Arabic numbers (0-9) using the information obtained from Kinect V2 (2.0 version). First, hand gesture features representing hand motion dynamics, including spatial position and direction of fingertips, are derived from Kinect. Then, the RBF neural networks are used to approximate hand motion dynamics underlying motion patterns of different gestures that represent Arabic numbers (0-9). Finally, the difference of motion dynamics will be applied to distinguish different hand gestures representing Arabic numbers. The outline of the proposed method has been shown in Fig. 1.

2.1 Data measurement and feature extraction

Our study uses hand gestures sequences captured by Microsoft Kinect V2 which can provide a 3D virtual skeleton model of the human body. The virtual skeleton consists of the positions of 25 joints and body parts, including the fingertips. Each user writes the Arabic numbers (0-9) with their right hand tip (since the participants are all right-handed). The 3D coordinate (x, y, z) value of each sign is captured by Kinect and the corresponding hand gesture is represented by the motion trajectory of the hand tip. Kinect provides approximately 30 skeleton frames per second. For each frame, the x, y and z coordinates are reported. A cer-

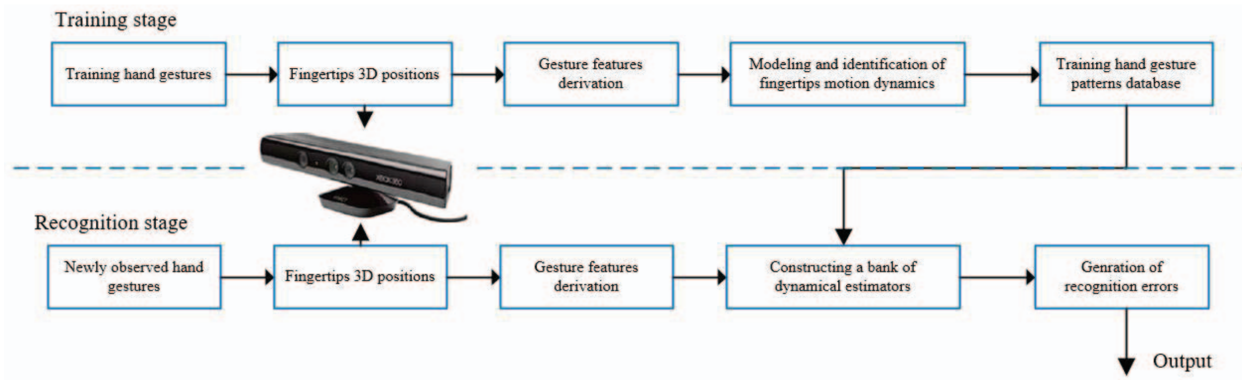

Fig. 1: Block diagram of the proposed method for hand gesture recognition with Kinect.

tain subject with his skeleton sample and the corresponding gesture are shown in Fig. 2.

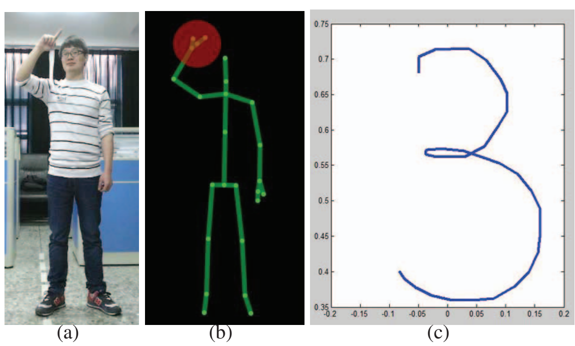

Fig. 2: Example of hand gesture data and its skeleton sample. (a) The person who is writing Arabic number 3; (b) the corresponding skeleton through Kinect; and (c) the corresponding hand gesture Arabic number 3.

Let a 3D spatio-temporal trajectory of the right hand tip be presented as a function of t:

$$\mathbf{D}(t) = [x(t), y(t), z(t)] \tag{1}$$

which is assumed to be differentiable on t. The hand motion trajectory derived from a practical motion capture system consists of a sequence of position vectors, $\mathbf{D}(t) = [x(t), y(t), z(t)]$, with t as the discrete-time index $0 \leq t \leq T$. $\mathbb{D}$, denoting the hand motion trajectory, can be written as $\mathbb{D} = [\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_T]$.

Then, we define the moving angles of the right hand tip in the $x-y$ plane and $x-z$ plane between $\mathbf{D}(t-dt)$ and $\mathbf{D}(t)$ as

$$\theta(t) = tan^{-1}(\frac{y(t)-y(t-dt)}{x(t)-x(t-dt)}), \quad -\pi < \theta(t) < \pi, \quad \forall t \tag{2}$$

and

$$\vartheta(t) = tan^{-1}(\frac{z(t)-z(t-dt)}{x(t)-x(t-dt)}), \quad -\pi < \vartheta(t) < \pi, \quad \forall t \tag{3}$$

where dt denote a time-infinitesimal.

2.2 Feature extraction and selection

For every time series of each hand gesture, we extract four different features to create our feature vectors for the recognition of hand gestures within $\mathbb{D}$. These features, including the $x(t), y(t)$ coordinates of the hand tip and the $\theta(t), \vartheta(t)$ angles of the hand tip, are employed to create a feature vector in four dimensions for each subject. Fig. 3 depicts the extracted four features of hand gestures through Kinect.

2.3 Training and learning mechanism based on selected features

In this section, we present a scheme for modeling and identification of motion dynamics of hand gestures which represent Arabic numbers (0-9) using the above mentioned gesture features.

In order to more accurately describe the hand gestures, hand motion dynamics can be modeled in the following form:

$$\dot{x} = F(x;p) + v(x;p) \tag{4}$$

where $x = [x_1, \ldots, x_n]^T \in R^n$ are the states of system (4) which represent the gesture features (including $x(t), y(t), \theta(t), \vartheta(t)$), p is a constant vector of system parameters. $F(x;p) = [f_1(x;p), \ldots, f_n(x;p)]^T$ is a smooth but unknown nonlinear vector representing the motion dynamics of hand gesture, $v(x;p)$ is the modeling uncertainty. The system trajectory starting from initial condition x_0, is denoted as $\varphi_\zeta(x_0)$.

Since the modeling uncertainty $v(x;p)$ and the motion dynamics of hand gesture system $F(x;p)$ cannot be decoupled from each other, we consider the two terms together as an undivided term, and define $\phi(x;p) := F(x;p) + v(x;p)$ as the motion dynamics of hand gesture system. The objective of the training or learning phase is to identify or approximate the general motion dynamics of hand gesture system $\phi(x;p)$ to a desired accuracy via deterministic learning.

Based on deterministic learning theory, the following dynamical RBF neural networks are employed to identify the motion dynamics of hand gesture system $\phi(x;p) = [\phi_1(x;p), \ldots, \phi_n(x;p)]^T$:

$$\dot{\hat{x}} = -A(\hat{x} - x) + \hat{W}^T S(x) \tag{5}$$

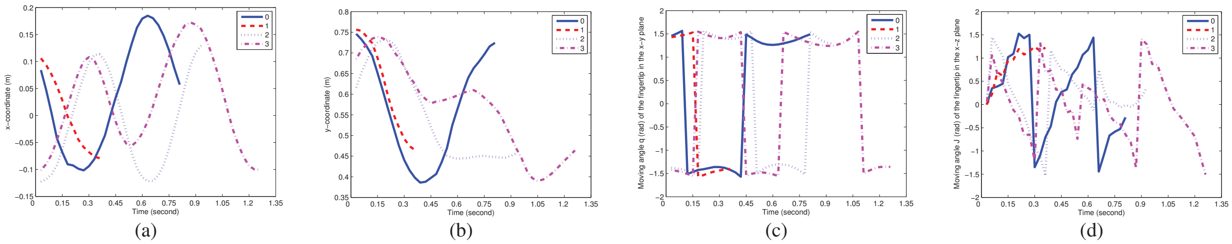

Fig. 3: The sample features of $x(t), y(t), \theta(t), \vartheta(t)$ for the gestures representing Arabic numbers 0, 1, 2 and 3.

where $\hat{x} = [\hat{x}_1, \ldots, \hat{x}_n]$ is the state vector of the dynamical RBF neural networks, $A = diag[a_1, \ldots, a_n]$ is a diagonal matrix, with $a_i > 0$ being design constants, localized RBF neural networks $\hat{W}^T S(x) = [\hat{W}_1^T S_1(x), \ldots, \hat{W}_n^T S_n(x)]^T$ are used to approximate the unknown $\phi(x; p)$. The employment of RBF NN is due to its associated properties, including the function approximation ability, the spatially localized structure and a property concerning the PE condition.

The NN weight updating law is given by:

$$\dot{\hat{W}}_i = \dot{\tilde{W}}_i = -\Gamma_i S(x)\tilde{x}_i - \sigma_i \Gamma_i \hat{W}_i \tag{6}$$

where $\tilde{x}_i = \hat{x}_i - x_i, \tilde{W}_i = \hat{W}_i - W_i^*$, W_i^* is the ideal constant weight vector such that $\phi_i(x; p) = W_i^{*T} S(x) + \epsilon_i(x)$, $\epsilon_i(x) < \epsilon^*$ is the NN approximation error, $\Gamma_i = \Gamma_i^T > 0$, and $\sigma_i > 0$ is a small value.

With Eqs. (4)-(5), the derivative of the state estimation error $\tilde{x}_i$ satisfies

$$\dot{\tilde{x}}_i = -a_i\tilde{x}_i + \hat{W}_i^T S(x) - \phi_i(x; p) = -a_i\tilde{x}_i + \tilde{W}_i^T S(x) - \epsilon_i \tag{7}$$

By using the local approximation property of RBF neural networks, the overall system consisting of dynamical model (7) and the NN weight updating law (6) can be summarized into the following form in the region Ω_ζ

$$\begin{aligned} \begin{bmatrix} \dot{\tilde{x}}_i \\ \dot{\tilde{W}}_{\zeta i} \end{bmatrix} &= \begin{bmatrix} -a_i & S_{\zeta i}(x)^T \\ -\Gamma_{\zeta i} S_{\zeta i}(x) & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{W}_{\zeta i} \end{bmatrix} \\ &+ \begin{bmatrix} -\epsilon_{\zeta i} \\ -\sigma_i \Gamma_{\zeta i} \hat{W}_{\zeta i} \end{bmatrix} \end{aligned} \tag{8}$$

and

$$\dot{\hat{W}}_{\bar{\zeta} i} = \dot{\tilde{W}}_{\bar{\zeta} i} = -\Gamma_{\bar{\zeta} i} S_{\bar{\zeta} i}(x)\tilde{x}_i - \sigma_i \Gamma_{\bar{\zeta} i} \hat{W}_{\bar{\zeta} i} \tag{9}$$

where $\epsilon_{\zeta i} = \epsilon_i - \tilde{W}_{\bar{\zeta} i}^T S_{\bar{\zeta}}(x)$. The subscripts $(\cdot)_\zeta$ and $(\cdot)_{\bar{\zeta}}$ are used to stand for terms related to the regions close to and far away from the trajectory $\varphi_\zeta(x_0)$. The region close to the trajectory is defined as $\Omega_\zeta := Z|\text{dist}(Z, \varphi_\zeta) \le d_\iota$, where $Z = x, d_\iota > 0$ is a constant satisfying $s(d_\iota) > \iota$, $s(\cdot)$ is the RBF used in the network, ι is a small positive constant. The related subvectors are given as: $S_\zeta(x) = [s_{j1}(x), \ldots, s_{j\zeta}(x)]^T \in R^{N_\zeta}$, with the neurons centered in the local region Ω_ζ, and $W_\zeta^* = [w_{j1}^*, \ldots, w_{j\zeta}^*]^T \in R^{N_\zeta}$ is the corresponding weight subvector, with $N_\zeta < N$. For localized RBF neural networks, $|\tilde{W}_{\bar{\zeta} i}^T S_{\bar{\zeta}}(x)|$ is small, so $\epsilon_{\zeta i} = O(\epsilon_i)$.

The nominal part of system (8) is referred to as system (8) without the terms $-\epsilon_{\zeta i}$ and $-\sigma_i \Gamma_{\zeta i} \hat{W}_{\zeta i}$. In Subsection 2.2, we have shown that the selected gesture features are quasi-periodic signals. Hence, the NN input $x = [x(t), y(t), \theta(t), \vartheta(t)]^T$ is quasi-periodic.

According to Theorem 1 in [11], the regression subvector $S_{\zeta i}(x)$ satisfies PE condition almost always. This will lead to exponential stability of $(\tilde{x}_i, \tilde{W}_{\zeta i}) = 0$ of the nominal part of system (8) [12]. Based on the analysis results given in [11], the NN weight estimate error $\tilde{W}_{\zeta i}$ converges to small neighborhoods of zero, with the sizes of the neighborhoods being determined by $\epsilon_{\zeta i}$ and $\|\sigma_i \Gamma_{\zeta i} W_{\zeta i}^*\|$, both of which are small values. This means that the entire RBF network $\hat{W}_i^T S(x)$ can approximate the unknown $\phi_i(x; p)$ along the trajectory φ_ζ, and

$$\phi_i(x; p) = \hat{W}_i^T S(x) + \epsilon_{i1} \tag{10}$$

where $\epsilon_{i1} = O(\epsilon_{\zeta i})$.

By the convergence result, we can obtain a constant vector of neural weights according to

$$\bar{W}_i = mean_{t \in [t_a, t_b]} \hat{W}_i(t) \tag{11}$$

where $t_b > t_a > 0$ represent a time segment after the transient process. Therefore, we conclude that accurate identification of the function $\phi_i(x; p)$ is obtained along the trajectory $\varphi_\zeta(x_0)$ by using $\bar{W}_i^T S_i(x)$, i.e.,

$$\phi_i(x; p) = \bar{W}_i^T S(x) + \epsilon_{i2} \tag{12}$$

where $\epsilon_{i2} = O(\epsilon_{i1})$ and subsequently $\epsilon_{i2} = O(\epsilon^*)$.

2.4 Recognition mechanism

In this section, we present a scheme for the recognition of hand gestures using the learned hand motion dynamics.

Consider a training dataset containing hand gesture patterns φ_ζ^k which represent Arabic numbers (0-9), $k = 1, \ldots, M$, with the kth hand gesture training pattern φ_ζ^k generated from

$$\dot{x} = F^k(x; p^k) + v^k(x; p^k), \quad x(t_0) = x_{\zeta 0} \tag{13}$$

where $F^k(x; p^k)$ denotes the motion dynamics of hand gesture system, $v^k(x; p^k)$ denotes the modeling uncertainty, p^k is the system parameter vector.

As shown in Subsection 2.3, the general motion dynamics of hand gesture system $\phi^k(x; p^k) := F^k(x; p^k) + v^k(x; p^k)$ can be accurately identified and stored in constant RBF neural networks $\bar{W}^{k^T} S(x)$. By utilizing the learned knowledge obtained in the training phase, a bank of M estimators is first constructed for the trained hand gesture systems as follows:

$$\dot{\bar{\chi}}^k = -B(\bar{\chi}^k - x) + \bar{W}^{k^T} S(x) \tag{14}$$

where $k = 1, \ldots, M$ is used to stand for the kth estimator, $\bar{\chi}^k = [\bar{\chi}_1^k, \ldots, \bar{\chi}_n^k]^T$ is the state of the estimator, $B = diag[b_1, \ldots, b_n]$ is a diagonal matrix which is kept the same for all estimators, x is the state of an input test pattern generated from Eq. (4).

In the recognition phase, by comparing the test hand gesture pattern (standing for a certain Arabic number) generated from the hand gesture system (4) with the set of M estimators (14), we obtain the following recognition error systems:

$$\begin{aligned} \dot{\tilde{\chi}}_i^k &= -b_i \tilde{\chi}_i^k + \bar{W}_i^{k^T} S_i(x) - \phi_i(x; p), \\ & \quad i = 1, \ldots, n, \quad k = 1, \ldots, M \end{aligned} \tag{15}$$

where $\tilde{\chi}_i^k = \bar{\chi}_i^k - x_i$ is the state estimation (or synchronization) error. We compute the average L_1 norm of the error $\tilde{\chi}_i^k(t)$

$$\|\tilde{\chi}_i^k(t)\|_1 = \frac{1}{\mathrm{T}_c} \int_{t-\mathrm{T}_c}^{t} |\tilde{\chi}_i^k(\tau)| d\tau, \quad t \geq \mathrm{T}_c \tag{16}$$

where T_c is the cycle of one certain hand gesture.

The fundamental idea of the recognition of hand gesture is that if a test hand gesture pattern is similar to the trained hand gesture pattern s ($s \in \{1, \ldots, k\}$), the constant RBF network $\bar{W}_i^{s^T} S_i(x)$ embedded in the matched estimator s will quickly recall the learned knowledge by providing accurate approximation to hand motion dynamics. Thus, the corresponding error $\|\tilde{\chi}_i^s(t)\|_1$ will become the smallest among all the errors $\|\tilde{\chi}_i^k(t)\|_1$. Based on the smallest error principle, the appearing hand gesture pattern can be recognized. We have the following recognition scheme.

3 Experimental results

The task of performance evaluation is to recognize 10 numerical symbols that contain various shape trajectories. The results have been obtained and the set of actual hand gesture symbols is depicted in Fig. 4. A Kinect device has been used to acquire the data relative to the performed gestures.

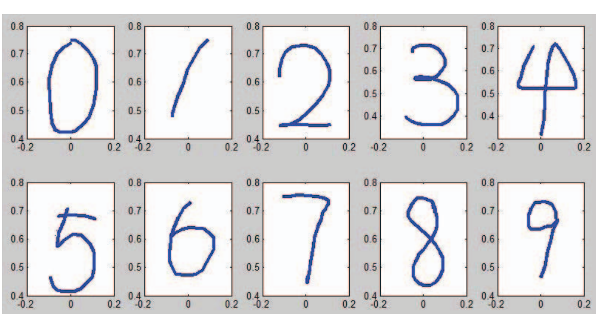

Fig. 4: Samples of gesture data extracted through Kinect.

The considered dataset of gestures contains 10 different gestures representing Arabic numbers (0-9) and is executed by 12 different persons. Each user has repeated each gesture 10 times for a total of 1200 different data samples. In order to compute the classification results we split the dataset in a training and a test set by using the cross-validation approach i.e., the 2-fold cross-validation and 10-fold cross-validation. For the 2-fold cross-validation, the dataset is divided into 2 subsets. In each subset, there are 60 data samples for each Arabic number, which including 600 data samples. Each time, one of the subsets is used as the test set and the other subset is used as a training set. For the 10-fold cross-validation, the dataset is divided into 10 subsets. In every subset, there are 12 data samples for each Arabic number, which including 120 data samples. Each time, one of the 10 subsets is used as the test set and the rest subsets are put together to form a training set. For the recognition of Arabic numbers (0-9), the total number of the samples are $10 \times 12 \times 10 = 1200$. By using the cross-validation, the gallery and probe sizes are shown in Table 1.

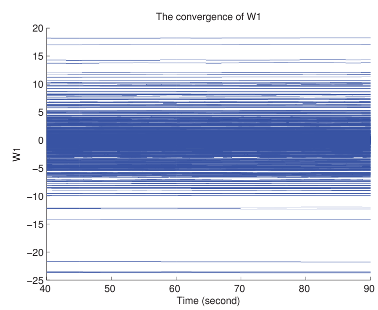

Fig. 5: Partial parameter convergence of $\hat{W}_1$ in one training pattern.

We assign feature vector sequences for all the 12 subjects with the gestures 0-9. Based on the method described in Subsection 2.2, we extract all the 12 subjects' gesture features through time series, which means the input of the RBF neural networks $x = [x(t), y(t), \theta(t), \vartheta(t)]^T$. In order to eliminate data difference between different gesture features, all the gesture feature data are normalized to $[-1, 1]$. Fig. 5 shows a training example of the gesture 0 by using 2-fold cross-validation method. The RBF network $\hat{W}_i^T S_i(x)$ is constructed in a regular lattice, with nodes $N = 83521$, the centers μ_i evenly spaced on $[-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1]$, and the widths $\eta = 0.15$. The weights of the RBF neural networks are updated according to Eq. (6). The initial weights $\hat{W}_i(0) = 0$. The design parameters for (5) and (6) are $a_i = 0.5, \Gamma = diag\{1.5, 1.5, 1.5, 1.5\}, \sigma_i = 10, (i = 1, \ldots, 4)$. The convergence of neural weights is shown in Fig. 5, which demonstrates partial parameter convergence.

The recognition process is carried out based on the method mentioned in Subsection 2.4. For the example of recognition of gesture 0 with 2-fold cross-validation method, by using the constant networks $\bar{W}_i^{k^T} S_i(x), (k = 1, \ldots, 600)$, 600 RBF network estimators are constructed based on (15). The parameter is $b_i = -25$ $(i = 1, \ldots, 4)$. Table 2 and

Table 1: Experiments on the self-constructed Kinect dataset for hand gesture recognition.

Hand Gestures	Cross-Validation Type	Gallery Size	Probe Size
Arabic numbers (0-9)	2-fold	600	600
Arabic numbers (0-9)	10-fold	1080	120

Table 2: Confusion matrix of hand gesture recognition on the Arabic numbers by using 2-fold method.

	0	1	2	3	4	5	6	7	8	9
0	57			1			2			
1		51	2		1	1	1	1		3
2			60							
3			1	59						
4					57					3
5				1		59				
6	3						57			
7				1		1		58		
8									60	
9					3					57

Table 3: Confusion matrix of hand gesture recognition on the Arabic numbers by using 10-fold method.

	0	1	2	3	4	5	6	7	8	9
0	117			1			2			
1		109	1		3		1	2		4
2			120							
3				119		1				
4					118					2
5						118			1	1
6	3						117			
7				3				117		
8									119	1
9		1			2				4	113

Table 4: Recognition results (%) on the Arabic numbers by using 2-fold cross-validation method.

Gesture	Gallery Number	Probe Number	Unsuccessfully Recognized Number	Recognition Rate (%)
0	600	60	3	95
1	600	60	9	95
2	600	60	0	100
3	600	60	1	98.33
4	600	60	3	95
5	600	60	1	98.33
6	600	60	3	95
7	600	60	2	96.67
8	600	60	0	100
9	600	60	3	95
			Average	95.83

Table 5: Recognition results (%) on the Arabic numbers by using 10-fold cross-validation method.

Gesture	Gallery Number	Probe Number	Unsuccessfully Recognized Number	Recognition Rate (%)
0	1080	120	3	97.5
1	1080	120	11	90.83
2	1080	120	0	100
3	1080	120	1	99.17
4	1080	120	2	98.33
5	1080	120	2	98.33
6	1080	120	3	97.5
7	1080	120	3	97.5
8	1080	120	1	99.17
9	1080	120	7	94.17
			Average	97.25

Table 3 show the confusion matrix of hand gesture recognition on the Arabic numbers by using 2-fold and 10-fold cross-validation methods, respectively. Table 4 and Table 5 show the recognition results on the Arablic numbers by using 2-fold and 10-fold cross-validation methods, respectively.

4 Conclusion

Our study contributes the accuracy improvements to the recognition of Arabic numbers (0-9) with the employment of Kinect. By using the 2-fold and 10-fold cross-validation styles, the correct classification rates are reported to be 95.83% and 97.25%, respectively. Overall, our classification approach achieves good performance, which indicates that the proposed system can be effective for the recognition of hand gestures.

REFERENCES

[1] Y. Zhou, G. Jiang, Y. Lin, A novel finger and hand pose estimation technique for real-time hand gesture recognition, Pattern Recognition, Vol. 49, 102-114, 2016.

[2] H. I. Suk, B. K. Sin, S. W. Lee, Hand gesture recognition based on dynamic Bayesian network framework, Pattern Recognition, Vol. 43, No. 9, 3059-3072, 2010.

[3] X. Yang, Y. L. Tian, Effective 3d action recognition using eigenjoints, Journal of Visual Communication and Image Representation, Vol. 25, No. 1, 2-11, 2014.

[4] G. Lu, Y. Zhou, X. Li, M. Kudo, Efficient action recognition via local position offset of 3D skeletal body joints, Multimedia Tools and Applications, Vol. 75, No. 6, 3479-3494, 2016.

[5] C. Zhang, Y. Tian, Histogram of 3D facets: A depth descriptor for human action and hand gesture recognition, Computer Vision and Image Understanding, Vol. 139, 29-39, 2015.

[6] A.I. Maqueda, C. R. del-Blanco, F. Jaureguizar, N. Garcła N, Human-computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns, Computer Vision and Image Understanding, Vol. 141, 126-137, 2015.

[7] J. Han, L. Shao, D. Xu, J. Shotton, Enhanced computer vision with microsoft kinect sensor: A review, IEEE transactions on cybernetics, Vol. 43, No. 5, 1318-1334, 2013.

[8] W. Lu, Z. Tong, J. Chu, Dynamic hand gesture recognition with Leap Motion controller, IEEE Signal Processing Letters, Vol. 23, No. 9, 1188-1192, 2016.

[9] R. Ibañez, Á. Soria, A. Teyseyre, G. Rodréguez, M. Campo, Approximate string matching: A lightweight approach to recognize gestures with Kinect, Pattern Recognition, Vol. 62, 73-86, 2017.

[10] H. Cheng, Z. Dai, Z. Liu, Y. Zhao, An image-to-class dynamic time warping approach for both 3D static and trajectory hand gesture recognition, Pattern Recognition, Vol. 55, 137-147, 2016.

[11] C. Wang, T. Chen, G. Chen, D.J. Hill, Deterministic learning of nonlinear dynamical systems, International Journal of Bifurcation Chaos, Vol. 19, No. 4, 1307-1328, 2009.

[12] J. Farrell, Stability and approximator convergence in nonparametric nonlinear adaptive control, IEEE Transactions on Neural Networks, Vol. 9, No. 5, 1008-1020, 1998.