International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 32, No. 6 (2018) 1850018 (14 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218001418500180

World Scientific
www.worldscientific.com

Sign Language Fingerspelling Recognition Using Depth Information and Deep Belief Networks

Yong Hu[*,†,‡], Hai-Feng Zhao[†] and Zhi-Gang Wang[†]

[*]*The Laboratory for Internet of Things and Mobile Internet*
Technology of Jiangsu Province
Huaiyin Institute of Technology, P. R. China

[†]*School of Software Engineering*
Jinling Institute of Technology, Nanjing, P. R. China

[‡]*huyong@jit.edu.cn*

Received 30 June 2017
Accepted 18 September 2017
Published 5 December 2017

In the sign language fingerspelling scheme, letters in the alphabet are presented by a distinctive finger shape or movement. The presented work is conducted for autokinetic translating fingerspelling signs to text. A recognition framework by using intensity and depth information is proposed and compared with some distinguished works. Histogram of Oriented Gradients (HOG) and Zernike moments are used as discriminative features due to their simplicity and good performance. A Deep Belief Network (DBN) composed of three Restricted Boltzmann Machines (RBMs) is used as a classifier. Experiments are executed on a challenging database, which consists of 120,000 pictures representing 24 alphabet letters over five different users. The proposed approach obtained higher average accuracy, outperforming all other methods. This indicates the effectiveness and the abilities of the proposed framework.

Keywords: Sign language fingerspelling recognition; depth information; deep belief networks; histograms of oriented gradients; Zernike moments.

1. Introduction

Sign Language Recognition (SLR) began in 1990s and has become one of the increasingly interested research areas in human–computer interaction domain. It aims at providing an efficient and accurate mechanism to translate sign language into text or speech.

Over the past 20 years, many techniques were presented to ameliorate the recognition rate and speed of sign language recognition. Many existing surveys were conducted to measure the development of this field.[1,8,14] In general, there are three

[‡]Corresponding author.

main categories in sign language recognition: sensor-based approach (using electromechanical devices such as data gloves), vision-based approach and the hybrid systems. Instrumented glove approaches simplified the recognition by measuring the different gesture parameters such as hand's position, angle, and the location of the fingertips. But data gloves are expensive, less user friendly and complicate users' interaction. The vision-based approaches did not require complex devices, only image processing techniques and bare hands. This enabled a natural interaction between users and system. Thus, it is widely used because of its characteristic of user friendly and affordable. The mixed systems obtained both information from visual devices and electromechanical instruments.

In American Sign Language (ASL), gestures are considered to be composed of two parts, static and dynamic. Static gestures are presented by a distinctive hand shape, and dynamic gestures are presented in succession hand movements. In sign language fingerspelling scheme, letters in the alphabet are presented by a distinctive finger shape or movement. It is a constrained but still an important part of ASL. In sign language fingerspelling, word is presented as a sequence of hand shapes or trajectories corresponding to individual letters. The visual similarity of some signs, the invisibility of the finger(s) and the large amount of variation by different signers all make the hand shape recognition very challenging.

In recent years, depth cameras are increasingly favored in machine vision and pattern recognition domain. As one of the popular depth cameras, Microsoft Kinect$^{\text{TM}}$ has been widely used as the image sensor. By using Kinect, RGB and depth information could be gathered to achieve a better recognition result.[5] Pugeault *et al.*[10] utilized Kinect for collecting intensity and depth images. In their work, Gabor filters and Random Forest are conducted for recognizing fingerspelling signs. The datasets they collected are also being adopted by some approaches for training and recognition. Uebersax *et al.*[13] proposed a framework for fingerspelling letters recognition in real-time. Segmentation and orientation estimation of hand area were conducted by using depth images. Zhu *et al.*[16] present an elastic fusion method for static sign language recognition. Common patch-level features were first extracted from intensity and depth information and then fused by means of kernel descriptors. Otiano Rodriguez *et al.*[7] proposed a kernel descriptor for fingerspelling letters recognition by using intensity and depth images. The Scale-Invariant Feature Transform (SIFT) and the Support Vector Machine (SVM) classifier are used in the approach. Lucas Rioux-Maldague *et al.*[11] proposed a novel feature extraction algorithm from depth and intensity images. Deep Belief Network (DBN) with three layers is utilized for classification. In the research of Yang,[15] hierarchical Conditional Random Field (CRF) and depth information were applied to recognize fingerspelling letters.

The presented work will concentrate on recognizing fingerspelling alphabet letters. A recognition framework by using intensity and depth information is proposed and compared with several existing works. Histogram of Oriented Gradients (HOG) and Zernike moments are used as discriminative features due to their simplicity and good performance. A DBN composed of three Restricted Boltzmann Machines (RBMs) is utilized for classification. Experiments are executed on a challenging database, which consists of 120 000 pictures representing 24 fingerspelling letters over five different users. The presented work is effective, simple to implement; and more than this, experimental results show better performance than the compared approaches. The arrangement of the remainder of the work is listed below. Background of materials and methods is introduced and detailed in Sec. 2. The proposed framework is presented in detail in Sec. 3. Section 4 shows the comparative experiments and results. The summary of the work and some suggestions are given in Sec. 5.

2. Background

2.1. *Segmentation of depth image*

Hand segmentation from the background is the prerequisite step of fusing the information from color and depth images. Many recognition systems depend on this critical step. Since the depth information has some advantages such as luminance independence, many segmentation techniques have been proposed based on depth information. There are mainly two ways to achieve segmentation threshold.

Most approaches use pre-defined or experimental threshold for segmentation.[9,10] In Pugeault' work[10]; hand area segmentation by using depth information is based on the assumption that the depth value of the region of interest (ROI) varies less than 20 cm in the depth dimension. In Lucas Rioux-Maldague's work,[11] the background is separated from the depth image by adopting a fixed threshold $\mathbf{t}$ on the maximum depth value. The depth $\mathbf{d}$ of the hand area is calculated by searching the minimum positive value in the depth matrix. Finally, pixels whose depth values are larger than $\mathbf{t}$ plus $\mathbf{d}$ are set to zero.

Other existing approaches deal with this problem in a different way. The ROIs was separated by using the clustering method.[7,9,16] The values of depth image are clustered into groups and then the closest cluster is labeled as foreground, owing to the fact that hand region is closer than other target in depth dimension. Prada[9] proposed a $\mathbf{k}$-means-based algorithm for performing the hand segmentation using depth values. At first, the depth values were gathered into $\mathbf{n}$ clusters and then sorted from the closest to farthest. Then, the closest cluster was labeled as the foreground. From the closest to the farthest, if the distance between the centers of consecutive clusters is less than a pre-defined distance $\mathbf{d}$, the next cluster is also labeled as foreground, otherwise the procedure terminates. The procedure depends on two parameters, $\mathbf{n}$ and $\mathbf{d}$, which are both pre-defined and experimental.

2.2. *Extract features*

In the finger spelling recognition field, the feature extraction step could be divided into three categories: extract from intensity images only, from depth images only, and from a mixed intensity-depth system.

Traditional recognition systems attempt to recognize finger alphabet from intensity images (color or gray). Since that hand detection and segmentation are critical steps in almost all of the vision-based methods, these 2D-based approaches have to face the difficulties of differentiating the hand from the background. When the background becomes more complex, the task of detection and segmentation becomes virtually impossible.

In recent years, depth information has generated considerable interest in the machine vision domain. Depth information was effectively applied in a lot of applications, including SLR systems. Depth cameras, such as Kinect or leap motion, provide a distance measurement of the observation object. By using depth information, it is much easier to detect and segment hand area form the background. Generally, depth information-based approaches are more reliable in hand detection and segmentation. Some recently proposed approaches achieved good performance based on depth images only.[7,13,16]

Most recently proposed approaches use both intensity and depth information for a better recognition rate. Mixed information always leads to more accurate result than intensity or depth information. Very convincing experimental results, including 54% recall rate and 75% precision rate on the database of more than 120,000 samples, were obtained in Pugeault's work[10] Zhu and Wong[16] achieved 89% precision on the same dataset by a fusion of common patch-level features of kernel descriptors. Otiniano *et al.*[7] achieved about 93% accuracy rate by using half samples for training, and 84% accuracy rate by using 10% of samples for training.

3. Proposed Recognition Framework

This part introduced and detailed the proposed framework for fingerspelling recognition by fusing intensity and depth information. Figure 1 illustrates the framework of the proposed SLR system. The proposed work is composed of three steps. The first step is called segmentation step, including depth image segmentation, gray conversion and filtering. In this step, an adaptive k-means-based segmentation approach is implemented for separating the hand area from the depth image. The segmentation

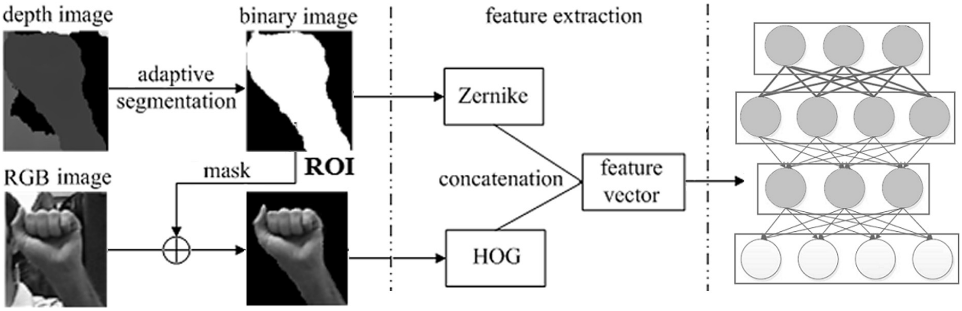


Fig. 1. Framework of the fingerspelling recognition system.

approach is fast and precise, without manual intervention. After the first step, the precise hand area is obtained for the next step — feature extraction. This step is composed of extracting discriminative features from intensity and depth images. HOG descriptors are applied to capture local shape information on intensity image, while Zernike moments are used to capture global shape information on depth image. In the third step, these features are combined to form a feature vector for classification. The obtained feature vector is then adopted for training a DBN classifier to obtain final results.

3.1. *Adaptive k-means based depth segmentation*

In the depth image, the depth value represents the real-world distance between object and sensor. Since the hand area is usually closer to the sensor, it means that the depth values of the hand region (foreground) are smaller than the values of the background. In some existing approaches, clustering procedures are adopted on depth information to obtain the threshold and then segment hand area from the background.[7,9,16] Unfortunately, they have the same shortcoming, depending on predefined and experimental parameters, such as cluster number **N**. To overcome this problem, we propose an adaptive depth segmentation algorithm based on **k**-means clustering. Figure 2 shows an example of the proposed algorithm. The presented procedure can effectively execute segmentation by using depth information, without manual intervention. The proposed procedure consists of five steps:

Step 1. Calculate the histogram of the depth image;

Step 2. From the histogram data, select **N** points as the initial cluster centroids, one for each cluster. The detailed process is as follows.

From small to large values, search continuous nonzero segments in the histogram data. Label the segment as a class and mark the value as the initial cluster centroids of this class (corresponding to the maximums). If the range of the data segment is too small, then merge the segment with the nearest adjacent segment (nearest first). In the depth image, the depth value represents the real-world distance between object and sensor. That means if the interval between two adjacent segments exceeds a certain value and the inside data are all zero, they must belong to different targets.

To simplify the clustering process, the upper limit of the class number **N** is pre-set as 5. The subsequent data will be pre-classified into one class when **N** reaches the upper limit. Since the depth value of the follow-up data is very large (means the target is not close to the camera), the foreground segmentation will not be substantively affected. After scanning the histogram data once, the cluster number **N** and the optimal initial cluster centroids are obtained automatically.

Step 3. Calculate the distances from cluster centers to every data point. Then associate the points to the nearest centroid according to the minimum distance. Usually, Euclidean distance is applied for calculating the distances.

Step 4. As the inclusion of new data points probably result in a transformation of the cluster centroids, all the class centers need to be recalculated after all the points are included in clusters.

Step 5. Set the termination condition: The centroids of the first two clusters are no longer changed or the total displacement is equal or less than a pre-defined number, or the upper limit iteration time is reached. The procedure terminates while the termination condition is met; or return back to step 3.

Since the hand area (foreground) is obviously the nearest in the depth image corresponding to the first cluster, the segmentation threshold can be easily obtained from the clustering results. After the segmentation was conducted on depth information by using a threshold, the ROI (binary hand area) is applied as a mask on RGB and depth images to split hand area from the background. Since the intensity and the depth images may not be lined up, an extra step must be executed before. Scaling and translation are conducted as a counterbalance for the position disparity of the Kinect sensors.

3.2. *Histograms of oriented gradients*

In recent years, HOG-based approaches has proved its efficiency on many applications. The HOG representation captures gradient structure that is very characteristic of local shape with an easily controllable degree of invariance to local geometric and photometric transformations. HOG descriptors operate on whole image and partition image window into cells with same size. Histogram of gradient orientations is calculated in every grid, and then concatenated to one feature vector. Two parameters could impact the generalization abilities of the feature, one is the size of the cells and the other is the interval size of the histograms. The diagram of rectangle HOG is shown in Fig. 3.

Based on the experiments in Refs. 2 and 12, HOG-based detectors greatly outperform the other popular shape features. Thippur *et al.*[12] valuated three frequently-used shape features about the efficiency and ability for sign language recognition. The evaluated features are HOG descriptors, Shape Context and Hu moments. Experimental results show that HOG descriptors achieved higher recognition rate for some signs. Navneet Dalal *et al.*[2] show experimentally grids of HOG descriptors significantly beyond Shape Context, wavelet and principle component analysis (PCA) plus SIFT. Moreover, the authors also discussed and analyzed the influence caused by varied HOG parameters.

HOG descriptor can be computed by the steps listed:

(1) Calculate gradients of the image

The gradients of the image can be obtained by using two one-dimensional filters: $[-1\ 0\ 1]$ for horizontal and $[-1\ 0\ 1]^{\mathrm{T}}$ for vertical.

(2) Partition the image into cells with the same size

The size of the cells was predefined in pixels and will impact the generalization abilities of these features. The blocks of cells could be overlapping or nonoverlapping.

(3) Compute the histogram of gradient of each cell

The histogram of gradient was computed by gathering votes to bins in every orientation. The greater number of bins will lead to more details of the histogram.

(4) Normalization

After the histogram of gradient was calculated, a normalization scheme is needed for each cell. The normalization factor could be obtained by L1-norm or L2-norm.

(5) Concatenation

The HOG descriptor of an image can be formed by combining every histogram into one feature vector.

3.3. *Zernike moments*

Moments-based features are frequently applied in SLR as global features. Zernike moments have the advantages of noise insensitivity, scale and rotation invariant. According to the experiments in Ref. 6, the Zernike moment features achieved a litter higher recognition rate than Hu moment features.

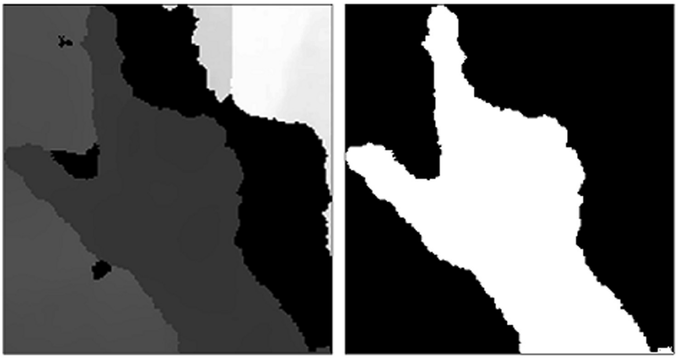


Fig. 2. Segmentation result from depth image.

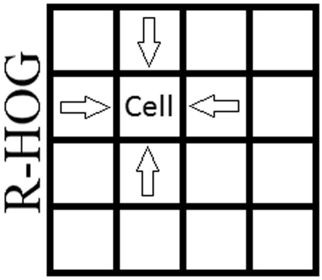


Fig. 3. Diagram of rectangle HOG.

For an intensity image $f(x, y)$, the Zernike moments with order p and repetition q are calculated by the formula:

$$Z_{pq} = \frac{p+1}{\lambda} \sum_{x} \sum_{y} f(x, y) V_{pq}^{*}(x, y), \tag{1}$$

where λ is a normalization factor and $x^2 + y^2 \leq 1$ means a unit circle.

By using the normalization technique, scale and translation invariance can be achieved. To obtain a normalized function $g(x, y, t)$ from image $f(x, y, t)$, a transform process must done first. The transforming formula is expressed as

$$g(x, y, t) = f\left(\frac{x}{a} + \overline{x}, \frac{y}{a} + \overline{y}\right). \tag{2}$$

3.4. *Deep belief networks*

Deep belief networks (DBNs) were first proposed by Hinton in 2006. The novel deep structured learning architecture was considered as a breakthrough on deep learning techniques. Due to the inherent capability of overcoming the drawback of traditional algorithms, deep learning techniques (include DBNs) have drawn ever-increasing research interests. Deep learning approaches have also been successfully applied to machine vision, human–computer interaction, speech recognition, and machine translation systems.[4]

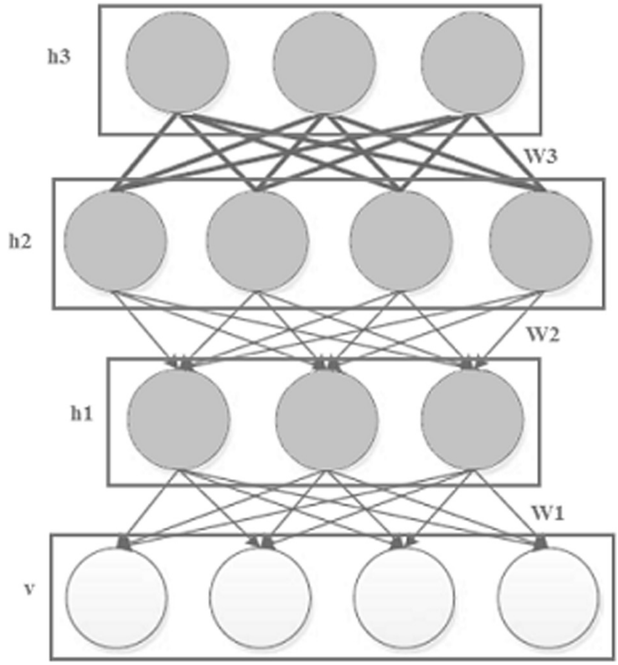


Fig. 4. Schematic diagram of DBNs with three hidden layers.

DBNs can be formed by the multilayer structure of Restricted Boltzmann Machines (RBMs) with disjointed visible and invisible layers. The composing of series of random binary cells makes the Boltzmann machine a coincident neural net. The network is composed of multiple invisible cells $h \in \{0,1\}^{g_h}$ and multiple visible cells $v \in \{0,1\}^{g_v}$. Invisible cells of each layer are disjointed to one another, so do visible cells. Superscripts g_v and g_h represent the amount of visible and invisible cells, respectively. Symbol $\{v,h\}$ represents the joint configuration of visible and invisible cells of Boltzmann machine. The energy can be calculated by the following equation.

$$E_{(v,h)} = -\frac{1}{2}(v^T Lv + h^T Jh) - v^T Wh. \tag{3}$$

In order to represent the energy $E_{(v,h)}$ in a simple way, the bias is taken away from the above equation. The variable W represents the concurrent weights amid visible and invisible cells. The variables L, J represent the concurrent weights among visible cells and invisible cells, respectively. For the variables L, J, values on the diagonal are all 0. Figure 4 represents the schematic diagram of DBNs with hidden layers.

4. Experiment and Analysis

4.1. *Dataset*

To measure the efficiency and capabilities of the presented framework, experiments are conducted on the widely used ASL Finger Spelling dataset built by Pugeault.[10] The dataset is a very challenging dataset in this domain, using diverse backgrounds and multiple users. It contains an easy set and a hard set, named as Datasets A and B, respectively. All samples are captured by using Microsoft Kinect TM sensors, with one RGB image and its homologous depth image. The dataset contains 24 of 26 alphabets except j and z, since both of them have motions, not static sign. Dataset A was represented by five users in constant illumination, while Dataset B was represented by nine users with illumination variations.

In Dataset A, each sign contains about 500 images generated by five users. The total number of RGB and depth images is 120,000. Figure 5 shows some examples of this dataset. The difficulties come from the fact that each user has its own specific way to represent the alphabet letters (the difference caused by different users can be seen in Fig. 5).

4.2. *Experimental setup*

Two experiments are constructed to test the presented approach. The adaptive k-means based depth segmentation algorithm is evaluated in the first experiment. In the second experiment, the performance of the proposed framework for finger spelling recognition is tested and analyzed.

For the experimental setup, the selection of the required parameters is detailed as follows. The interval threshold T is usually set to 50–200. Since the depth image

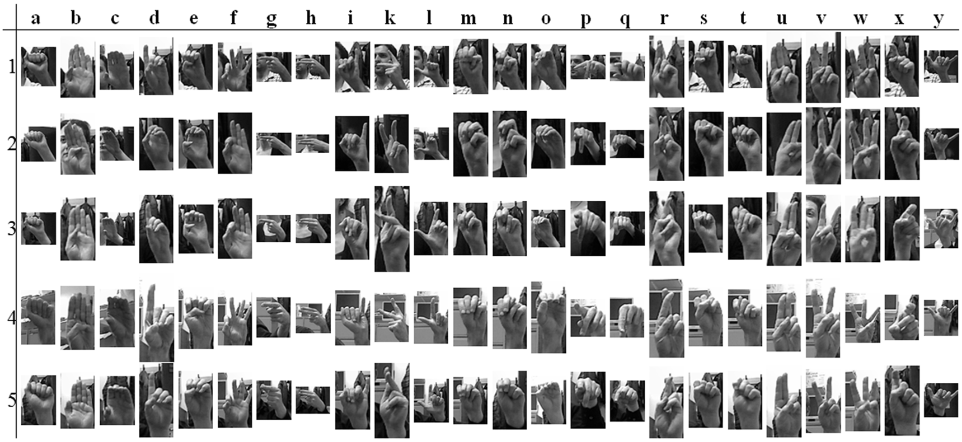


Fig. 5. Some examples of ASL fingerspelling dataset.

resolution captured by Kinect is 1 mm, the interval threshold T used in segmentation stage is set to 100, equal to 10 cm in real world distance. In extracting HOG features, local window was set to 3×3 and the number of histogram bins was set to 9. A feature vector of 81 dimensions was extracted from the finger spelling images. As to the Zernike moments, scale/translation normalized moments were calculated from the centered image. A feature vector of 15 dimensions was obtained by up to order 6. Note that only the real part was used in this work. A DBN composed of three RBMs is used as classifier. A powerful toolbox includes almost all the qualities required, DeeBNet,[3] are implemented in experimental analysis. The toolbox is object-oriented; running on MATLAB software. The toolbox has two packages with some classes and functions for managing data and sampling methods and also has some classes to define different RBMs and DBN.

4.3. *Results and analysis*

The first experiment was conducted to evaluate the proposed adaptive k-means based depth segmentation algorithm. Table 1 shows the percentage of cluster number **N** of depth images in database. It can be observed clearly that the number **N** varied from 2 to 5. That means the existed **k**-means based depth segmentation methods have to face the difficult of defining **N**. Eventually, it will lead to either misclassified or time consuming.

The distribution information of iteration times and total displacement is shown in Table 2. It is observed from the Table that the proposed approach only iterates once

Table 1. The percentage of cluster number N of depth images.

Cluster Number **N**	2	3	4	≥ 5	Average
Percentage	7.65%	45.28%	37.90%	9.17%	3.4824

Table 2. The distribution information of iteration times and total displacement.

Iteration Times	1	2	3	≥ 4	Average
Percentage	91.68%	3.97%	1.70%	2.65%	1.1785
Total displacement	0	(0, 0.05]	(0.05, 01]	> 0.1	Average
Percentage	98.49%	0.78%	0.71%	0.01%	0.0012

Table 3. Average accuracies and standard deviation (STD).

Train (%)	Test (%)	Accuracy (%)	STD
10	90	88.12	0.14
20	80	90.9	0.12
30	70	91.72	0.12
40	60	92.63	0.10
50	50	95.42	0.08

in most cases, and the average iteration times is much lower. It indicated that the initial cluster centroids are selected well. It can also be seen from the Table that the total displacement is much lower. Due to the well selected initial cluster centroids, the centroids do not change in most cases. Thus, the computation cost is reduced significantly. All in all, adaptive cluster number **N** and well selected initial cluster centroids are the features of the proposed approach. That makes the approach accurate, fast and less computation cost.

The second experiment was conducted to evaluate the proposed framework for finger spelling recognition. The experiment results were compared with the approaches proposed by Pugeault and Bowden,[10] Zhu and Wong[16] and Otiniano *et al.*[7] These approaches use both intensity and depth information on the same dataset.

To evaluate the robustness of the extracted features, varied percentages of training samples were applied in the experiment first. Table 3 shows the results of accuracies and average standard deviation (STD) from the proposed approach. We can see from the table that an accuracy rate 95.42% is achieved when half training samples are used, while an accuracy rate 88.12% is achieved when one-tenth of training samples are used.

Table 4 presents the classification rate and STD by using half samples for training. We can see from the table that the presented approach achieved the highest

Table 4. Accuracies and STD by using half samples for training.

Approach	Classification Rate (%)	STD
Pugeault[10]	75	—
Zhu[16]	89.23	0.14
Otiniano[7]	91.26	0.17
The proposed approach	94.37	0.09

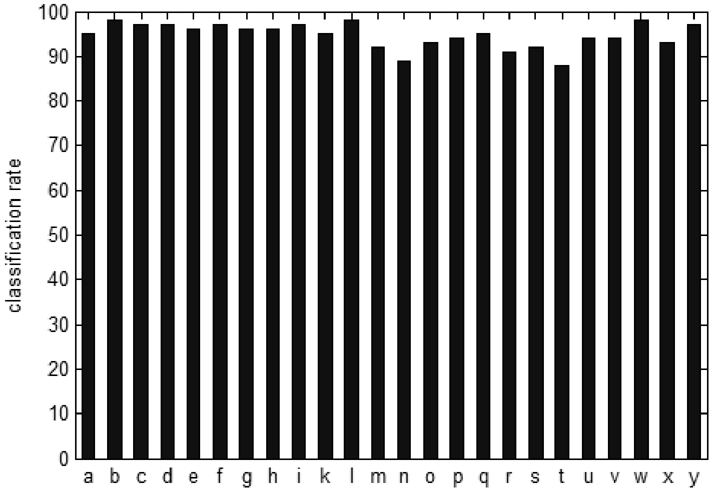


Fig. 6. Classification rate of each alphabet letter (%).

average accuracy, outperforming all other methods. The experimental results indicate the abilities and good performance of the presented framework.

Figure 6 shows the classification rate of each alphabet letter of the presented system. By using a combined feature vector, a high accuracy rate was obtained. More than half of the signs have the recognition rate of over 95%. In the meantime, there is some misclassification between similar fingerspelling signs. The signs *t* and *n* obtained minimum classification rate, 88% and 89%, respectively. There are two reasons behind this: one is that both alphabet letters are represented by a fist with different thumb positions, the other is that thumbs are almost invisible in some cases.

The performances of time consumption are also evaluated in the experiments. The system was implemented on a laptop (Intel Core$^{\text{TM}}$ i3-2310M CPU @ 2.1 GHz, 4 GB of RAM) running under 64 bit windows 7 and MATALAB 2009a. The procedure for each sample (include a depth image and a RGB image) requires 29.7 ms in average. The step of adaptive *k*-means-based depth segmentation requires approximately 8.6 ms. The step of feature extraction requires 17.9 ms, where 52% of the computation time is required to extract HOG. With these results, it is confirmed that the proposed system is fast enough to run in real time.

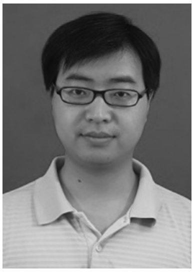
5. Summary and Conclusions

In the presented work, a recognition framework by fusing intensity and depth information was proposed and compared with some distinguished works. HOG and Zernike moments are used as discriminative features due to their simplicity and good performance. A DBN composed of three RBMs is used as classifier. Experiments are executed on a challenging database, which consists of 120,000 pictures. The proposed approach obtained higher average accuracy, outperforming all other methods. This indicates the effectiveness and the abilities of the proposed framework.

References

1. H. Badi, A survey on recent vision-based gesture recognition, *Intell. Ind. Syst.* **2** (2) (2016) 179–191.
2. N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, *Int. Conf. Computer Vision & Pattern Recognition (CVPR'05)*, 20–26 June 2005, Vol. 2, San Diego, CA, USA, pp. 886–893.
3. M. A. Keyvanrad and M. M. Homayounpour, A brief survey on deep belief networks and introducing a new object oriented toolbox (DeeBNet), arXiv:1408.3264v7 [cs.CV].
4. W. Liu *et al.*, A survey of deep neural network architectures and their applications, *Neurocomput.* **234** (2016) 11–26.
5. R. Lun and W. Zhao, A survey of applications and human motion recognition with microsoft kinect, *Int. J. Pattern Recognit. Artif. Intell* **29**(5) (2015) 1–49.
6. K. C. Otiniano-Rodríguez, G. Camara-Chavez and D. Menotti, Hu and Zernike moments for sign language recognition, in *2012 Int. Conf. Image Processing, Computer Vision, and Pattern Recognition (IPCV'12)*, 16–19 July 2012, Las Vegas, USA, pp. 1–5.
7. K. Otiniano-Rodriguez *et al.*, Finger spelling recognition using kernel descriptors and depth images, in *28th SIBGRAPI — Conf. Graphics, Patterns and Images* (Salvador, Bahia, Brazil, 2015), pp. 72–79.
8. P. K. Pisharady and M. Saerbeck, Recent methods and databases in vision-based hand gesture recognition: A review, *Comput. Vis. Image Underst.* **141**(C) (2015) 152–165.
9. F. Prada, L. Cruz and L. Velho, Improving object extraction with depth-based methods, in *XXXIX Latin American Computing Conf. (CLEI'2013)*, 7–11 October 2013, Club Puerto Azul, Naiguata, Venezuela, pp. 1–9.
10. N. Pugeault and R. Bowden, Spelling it out: Real-time ASL fingerspelling recognition, in *Proc. IEEE Int. Conf. Computer Vision Workshops (ICCV Workshops)*, 6–13 November 2011, Barcelona, Spain, pp. 1114–1119.
11. L. Rioux-Maldague and P. Giguere, Sign language fingerspelling classification from depth and color images using a deep belief network, in *2014 Canadian Conf. Computer and Robot Vision*, 6–9 May 2014, Montreal, QC, Canada, pp. 92–97.
12. A. Thippur, C. Henrik Ek and H. Kjellstrom, Inferring hand pose: A comparative study of visual shape features, in *10th IEEE Int. Conf. Workshops on Automatic Face and Gesture Recognition (FG'2013)*, 22–26 April 2013, Shanghai, China, pp. 1-8.
13. D. Uebersax *et al.*, Real-time sign language letter and word recognition from depth data, in *Proc. IEEE Int. Conf. Computer Vision Workshops* (*ICCV Workshops*, 2011), pp. 383–390.
14. U. Von Agris *et al.*, Recent developments in visual sign language recognition, *Univers. Access. Inf. Soc.* **6**(4) (2008) 323–362.
15. H. D. Yang, Sign language recognition with the kinect sensor based on conditional random fields, *Sensors*, **15**(1) (2015) 135–147.
16. X. Zhu and K.-Y. K. Wong, Single-frame hand gesture recognition using color and depth kernel descriptors, in *Proc. 21st IEEE Int. Conf. Pattern Recognition (ICPR)*, 11–15 November 2012, Tsukuba Science City, Japan, pp. 2989–2992.

Yong Hu received his Ph.D. from the School of Computer Science & Technology at Nanjing University of Science and Technology (NUST) in 2010. He is currently an Associate Professor at Jinling Institute of Technology (JIT). His main research interests include image processing, pattern recognition and machine learning.



Hai-Feng Zhao received his B.E. and Ph.D. degrees from Nanjing University of Science and Technology (NJUST), China, in 2005 and 2012, respectively. He is currently a Senior Engineer with the School of Software Engineering, Jinling Institute of Technology, China. Before that, he was an Assistant Researcher with the Shenzhen Institutes of Advanced Technology at the Chinese Academy of Sciences. He visited the Australian National University and Canberra Research Laboratory of NICTA as a visiting student from September 2008 to August 2010. His research interests include computer vision, pattern recognition and human computer interaction.

Zhi-Gang Wang received his masters degree of Engineering from Southeast University (SEU), China, in 2006. He is currently an Associate Professor at Jinling Institute of Technology (JIT). His main research interests include data mining and artificial intelligence.