Sparse Representation for Kinect Based Hand Gesture Recognition System

Zeke Xu, Zhenhao Huang, Zhuoxiong Zhao, Zhiyuan Li, Pengsen Huang

Electronic and Information Engineering, South China University of Technology

Abstract

Hand gesture recognition that has proven a significant factor to directly influence the nonverbal communication between human and computer is becoming a challenging topic in the field of machine vision. This paper aims to propose a novel hand gesture recognition system which applies sparse representation to the Kinect to improve the efficiency of Kinect-based human-computer interaction. Auto-encoder neural network computation is also utilized to achieve better result. The sparse auto-encoder neural network is versatile and robust in complex features learning and computational efficient. Finally, results indicate that our algorithm greatly facilitates the gesture recognition rate up to 95%.

Keywords: human-computer interaction, hand gesture recognition system, sparse representation, Kinect, auto-encoder neural network computation

1. Introduction

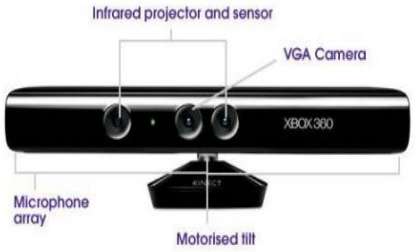
In recent years, the human-computer interaction has been challenged by more and more oncoming issues. Among them, the vision-based hand gesture recognition technology, an important part of human-computer interaction, has been attracting the concern of many researchers [1, 2]. In the past decades, there were many hand gesture recognition techniques developed for tracking and recognizing various hand gestures. Most of them require pretreatment of image features like texture, color etc. However, skin tones and texture change very rapidly from person to person and continent to continent. Further, different illumination conditions lead to modified color texture and ultimately change the observed results [3].

In order to avoid the problem above, we try to recognize different hand gestures in a statistical manner in our system. Natural images have essential statistical regularities that distinguish them from other kinds of input images [4]. Most of the statistical models have taken into account the non-Gaussian properties of static image patches, which lead to sparse coding and independent component analysis (ICA). These models [5, 6] built on natural image statistics are based on one particular statistical property-sparseness which can be used to simulate simple cell receptive field properties. In recent research reported by [7] Wright, John, sparse representation has proven an extremely powerful tool for acquisition, representation, and compression of multi-dimensional signals. However, the models of sparse coding and ICA is computationally costly either in testing time or in training time. Therefore, we employ the sparseness constraint to the auto-encoder neural network to balance the computational complexity and efficiency of gesture recognition. In order to further improve the recognition rate, we also separate the hand type from the surrounding environment using the skeleton tracking

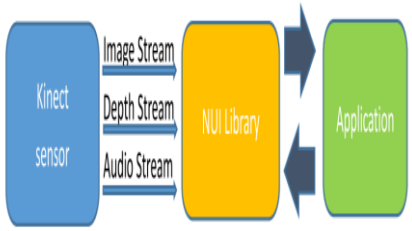
of Kinect before images modeling. Finally, we have achieved the high recognition rate up to 95%.

2. A brief introduction of Kinect

The Kinect (see Fig. 1) sensor, issued by Microsoft on June, 2010, contains a depth sensor, a RGB color camera, and a microphone array. Also, Kinect provides full-body 3D motion capture, and voice recognition capabilities [8].



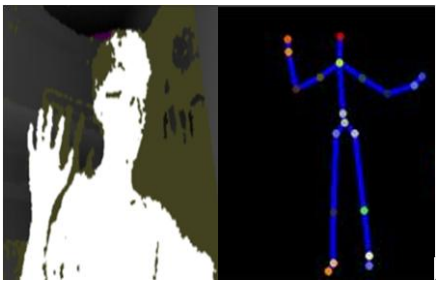
(a)



(b)

Fig. 1: (a) shows the physical map of the Kinect sensor; (b) shows how the sensor data stream flow

In this paper, we mainly focus on the depth sensor and skeleton tracking using Kinect. Fig. 2 shows the depth map and the skeleton map produced by the Kinect sensor. The depth value is encoded with gray values.



(a) (b)

Fig. 2: Depth map (a) and skeleton map (b)

3. Sparse auto-encoder neural network

Sparse representation, based on simulating human visual system, has been extensively studied in the image features extraction method, and shows that one single image can be accurately recovered by a sparse linear combination of the overall data. However, the traditional sparse representation algorithms have not done well. Sparse coding is computationally expensive to test and ICA takes much time to train. So we explore a new model of sparse feature representation in this paper. We combine the sparse representation with the auto-encoder neural networks to extract feature from patches.

Fig. 3 shows the architecture of auto-encoder neural network.

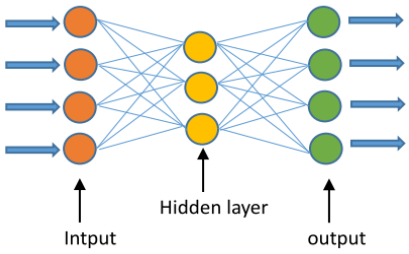


Fig. 3: Neural network model

From the Fig. 3, we can formulize the computation model of the neural network as [9]:

$$\begin{aligned} H(W, A, X) &= f(A \times f(W \times X)) \\ f(X) &= \frac{1}{1 + \exp(-X)} \\ A &= \left[A_1, A_2 \cdots, A_m\right] \\ W &= \left[W_1^T; W_2^T \cdots; W_m^T\right] \end{aligned} \tag{1}$$

A classic approach to reconstruct the image X is a linear weighted sum of features. Let us denote each feature by:

$$A_I \in R_{n \times 1}; i = 1,2\ldots, m \tag{2}$$

These features are assumed to be fixed. For each incoming image X, the coefficients of each feature in an image are denoted by the element of column vector S and each element value also represents the activation value of each hidden unit. Generally we defined the operation from hidden layer to output layer as data reconstruct and it can be formulized as:

$$X' = A \times S \tag{3}$$

However, for a given image X, how to compute the feature coefficient matrix S is the key point for reconstruction. In our paper, we can solve this problem easily by means of the operation from input layer to hidden layer. We define some feature detectors firstly denoted by:

$$W_i \in R_{n \times 1}; i = 1,2\ldots, m \tag{4}$$

Therefore the coefficient matrix S can be computed linearly as:

$$S = W \times X \tag{5}$$

But in our experiment, we will defined a non-linear function called activation function f to represent the non-linear properties of the neuron, which will help our training.

The auto-encoder network implies that the information transformed from the input to the output has been well captured and saved. Thus, we can readily regard the hidden activations as one representation of the input data.

In this paper, we use batch gradient descent algorithm [10] to train our sparse auto-encoder neural network. Details of the algorithm are shown below:

- Step1: set a cost function:

$$\begin{aligned} J &= \left\|X'-X\right\| + \lambda\left(\left\|A\right\| + \left\|W\right\|\right) + \beta\left\|S\right\| \\ \lambda &= 0.01,\ \beta = 0.1 \end{aligned} \tag{6}$$

  Hereinto, the first term is the reconstruction error term, the second term is a weight decay term that tends to decrease the weights and prevent over-fitting, and the last term represents the sparseness of hidden unit activation.
- Step2: calculating partial derivative of the cost function based on BP (back-propagation) algorithm [11].
- Step3: optimize the cost function to get the minimum using the batch gradient descent algorithm.

After training, we use the forward propagation to compute each input data to get the representation for one hidden layer, namely the sparse representation.

4. Hand gesture recognition algorithm

4.1. Hand gesture segmentation

Kinect is able to detect the depth of the entire scene within the scope of [85cm, 4000cm] and returns a 640×480 real time image and a corresponding RGB image through a pair of IR projector and camera. In our experiment, we use the Kinect to capture the depth data from the target scene where the target hand profile

should be clearly and easily captured. On the other hand, we can also get the skeleton map of the target hand as well as the center of hand palm. Once we get the depth data and skeleton map [12], the position of the hand palm can be easily detected and hand gesture segmentation can be perfectly carried out by the way of setting threshold (see Fig. 4).

Fig. 4: the leftmost map is the skeleton point of hand palm (palm center and wrist); the middle is the depth map; the rightmost is the hand palm after segmentation.

4.2. Train the sparse auto-encoder neural network

In this paper, we take patches (sized 10x10) from the hand gesture segmentation map for training because the map is too large to train the model. As we know, natural images have the property of being stationary [13], meaning that the statistic regularity of one part of the image is the same as the whole one. This also suggests that the features we within one part of the image can also be applied to other parts of the image, and we can use the same features throughout the whole image.

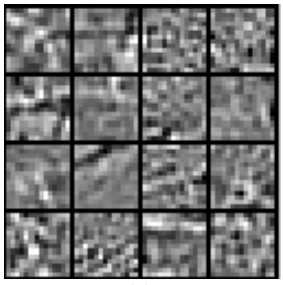
4.3. Convolution, pool and classify

After training, we can get feature detectors. Generally, we will use the feature detectors to detect features of all the patches within one single image. In another word, we make the feature detectors convolution with the large image [14]. In order to achieve invariant features, we also compute the features of adjacent patches which is termed pool [15]. In this paper, un-overlapped pooling is used. Finally, we use the Soft-max regression classifier [16] to recognize based on the pooled features.

5. Experiment

5.1. Compare sparse coding, ICA and sparse auto-encoder

In this experiment, we implement the ICA, sparse coding and sparse auto-encoder algorithm using the standard dataset (including 10 512x512 natural images) to compare their effect. In Fig. 5, we show the trained feature detectors of the three algorithms. From the Fig. 5, it implies that the sparse auto-encoder algorithm is able to learn the best feature detectors with the same training time (we set 10mins). In detailed description, the feature detectors of sparse auto-encoder are smoother and have distinct edges with different location and orientation. These detectors can be used to detect different features in one image. Therefore, our experiment result shows that the sparse auto-encoder really is less time-consuming



(a)

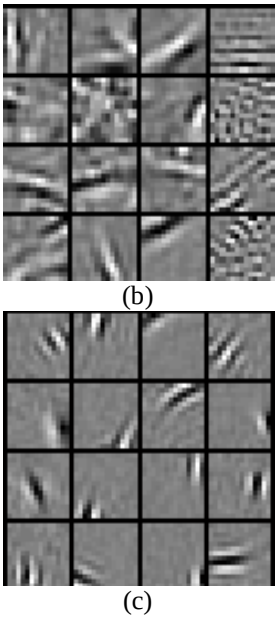


Fig. 5: Each represents 16 feature detectors of ICA (a), sparse coding (b) and sparse auto-encoder (c). According to (a), we can see that the result of the ICA shows random noise. And the result of sparse coding (b) implies some clear space filters but still not enough perfect. However, (c) shows the smoother filters and has abundant local directivities.

5.2. Hand gesture recognition

In this experiment, we collected 500 hand gesture segmentation maps from several persons using Kinect to train our recognition system. For convenience, there are five different categories (each category has 100 samples). We also sample other 500 for test. We set 100 input units and 400 hidden units. Also we set the pool region sized 3x3. In order to confirm the necessity of hand gesture segmentation, we also use the unprocessed samples. Finally we find that segmentation truly improve the recognition accuracy or rate by about 5%. Our system is trained with 8GB RAM and costs half an hour to achieve the recognition rate or accuracy up to 95%. In Table 1and Table 2, we also find that if we want to recognize more kinds of hand gestures, we should introduce more samples into our training system. To some extent, if the amount of feature detectors to be trained is increased, the result shows the recognition rate can be improved. However, if there are too many features, more samples are needed. Otherwise the accuracy or recognition rate will be reduced.

Table 1: contrast of different kinds of gesture

samples	features	gesture class	accuracy rate
500	400	5	95%
1000	400	10	92.3%
2000	400	20	90.0%

Table 2: contrast of different features

features	gesture class	accuracy rate
350	5	90%
400	5	95%
450	5	92%
500	5	89%

6. Conclusion

In this paper, we put forward a Kinect based sparse auto-encoder hand gesture recognition system and achieve the recognition rate up to 95%. It is notable that our system's performance degrades if the hand segmentation is removed. We has proven that removing the hand segmentation results in a loss of about 5% for the final performance of our system. Besides, we employ the sparse auto-encoder to simulate the human visual system and reduce the computational complexity. With concrete control experiment, we show that sparse auto-encoder algorithm behaves more satisfactory authentically.

As we know, human brain is absolutely an undeveloped research field to which more and more attentions need to be paid, but many achievements till now may be applicable to greatly facilitate our recognition. Therefore, we would like to introduce more effective properties of the primary visual cortex to our system in the future research work.

7. Acknowledgement

This work is supported by Science and Technology Planning Project of Guangdong Province of China (NO.2011A010801005,2010A080402015); National Undergraduate Innovative and Entrepreneurial Training Program of China (NO.20111056121).

8. References

[1] S. Mitra, and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, pp. 311-324, 2007.

[2] Y. Wu, and T. S. Huang, "Vision-based gesture recognition: A review," *Urbana* 51, 1999.

[3] P. Garg, N. Aggarwal, and S. Sofat, "Vision based hand gesture recognition," *World Academy of Science, Engineering and Technology*, pp. 972-977, 2009.

[4] A. Hyvärinen, "Statistical Models of Natural Images and Cortical Visual Representation," *Topics in Cognitive Science* 2.2, pp. 251-264, 2010.

[5] B. A. Olshausen, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, pp. 607-609, 1996.

[6] J. H. van Hateren, and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, pp. 359-366, 1998.

[7] J. Wright, Y. Ma, J. Mriral, G. Sapiro, T. S. Huang, and S. Yan "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, pp. 1031-1044, 2010.

[8] Z. Zhang, "Microsoft Kinect sensor and its effect," *Multimedia, IEEE*, pp. 4-10, 2012.

[9] M. T. Hagan, H. B. Demuth, and M. H. Beale, "Neural network design," *Boston London: Pws Pub*, 1996

[10] S. Boyd, and L. Vandenberghe, "Convex optimization," *Cambridge university press*, 2004.

[11] R. Hecht-Nielsen, "Theory of the backpropagation neural network," *Neural Networks, 1989. IJCNN., International Joint Conference on*. IEEE, 1989.

[12] D. Catuhe, "Programming with the Kinect for Windows Software Development Kit: Add Gesture and Posture Recognition to Your Applications," *O'Reilly Media, Inc.*, 2012.

[13] A. Hyvärinen, J. Hurri, and P. O. Hoyer, "Natural image statistics," *Vol. 39*. *Springer*, 2009.

[14] Y. LeCun, and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*3361, 1995.

[15] H. Schulz, and S. Behnke, "Learning object-class segmentation with convolutional neural networks," *11th European Symposium on Artificial Neural Networks (ESANN)*. Vol. 3. 2012.

[16] D. Heckerman, and C. Meek, "Models and selection criteria for regression and classification," *Proceedings of the thirteenth conference on uncertainty in artificial intelligence*. *Morgan Kaufmann Publishers Inc.*, 1997.