A Framework for Hand Gesture Recognition with Machine Learning Techniques

2013-20738, Chanseok Kang (Prof. Bernhard Egger)
2013-20760, Sangchul Kim (Prof. Srinivasa Rao Satti)

Abstract

Generally, people have difficulty on understanding specific gesture such as American Sign Language. To overcome this difficulty, the real-time gesture interpreter is required for gesture communication. However, there is no clue to understand the meaning of gesture through the computer directly. It needs to define the generalized model in the computer. That is why the machine-learning approach implements in recognition system. In this paper, we proposed the method of static hand gesture recognition in real time using multiclass Support Vector Machine (SVM). To train the model, we collect the 100 depth-based Histogram of Oriented Gradient (HOG) features per alphabet. From this dataset, we can generate the generalized gesture model for each alphabet image. The success rate of our proposed method is 92.8%.

Keywords
Hand Gesture Recognition, Histogram of Oriented Gradient, Support Vector Machine.

1 Introduction

In recent years, Human-Computer Interaction combined with computer vision is one of hot topics in computer-related research area. Above all, gesture recognition is vitally continuing study from now on. Unlike the verbal language, the gesture contains the meaning inside the visual context.

The sign languages are needed for special situation when people cannot talk each other. For example, it is only allowed hand gesture to communicate under the sea. Expert skin divers can interpret hand gestures immediately, but the beginners have difficulty on understanding that signals. Most people are not familiar with this kind of sign languages, so that interpreters are required. Although there are several approaches to gather the hand gesture information using 2D-camera, some limitations are existed in model-based and color-based approach. We can only use the color information (RGB, YCbCr) for classification, and its value depends on the view of observers.

In 2011, Microsoft invented the 3D-depth sensor named `Kinect`. It was the optional game device of XBOX for improving the game experience, but some geeks focused on the purpose of research. Unlike the previous camera sensors, Kinect can obtain depth data from current frames. Using this, we can convert from pixel information to 3D-real world features.

There are lots of research using features such as game, rehabilitation, 3D-reconstruction and so on. Unfortunately, it is difficult to beat the environment constraints. If we use the depth data for dataset, we don't care about the environment variable such as colors, and we can reduce the complex dimension for classification. That is why we choose the Kinect for hand gesture recognition.

Figure 1: 3D-depth sensor (Kinect)

2 Related Work

To recognize the hand gesture, hand detection techniques are required. In the previous research, there were several approaches to detect the hand part [1]. One of the approaches is hand modeling approach. Some use the wearable device to gather the 3D point of specific hand parts, and reconstruct them based on collected data [2]. Compared with other models, this approach is very accurate estimation, but if we want to consider the finger modeling, the target Degree-Of-Freedom (DOF) is too high. It is complex to get the perfect hand model. The other approach is a computer-vision based hand tracking [3]. Color-based methods are main stream for that [4]. However, hand motions are non-trivial task under lighting conditions. Some research use skin color as a tracking parameter. In these kinds of method, background segmentation is needed. There is an overhead of preprocessing for segmentation. To summarize it, color-based hand tracking is good approach, but it has some limits for implementation.

After appearance of stereo camera, some research use depth data instead of color [5, 6]. It can obtain the 3D points of hand without wearable device. Due to the low

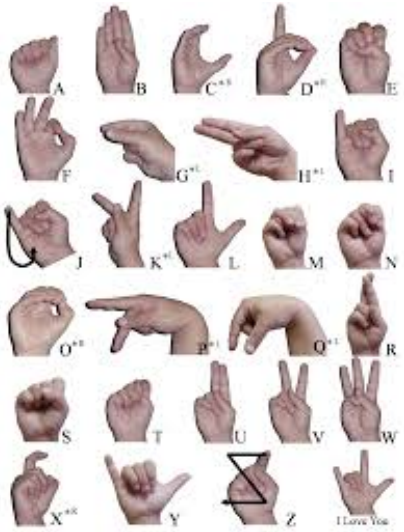


Figure 2: American sign language

resolution of stero camera, it can be only used in limited gesture. In 2005, Histogram of Oriented Gradients [7], the effective method for object detection, is announced. Some implement this in color-based tracking as a tracking parameter. It can reduce the defect of methods but not perfectly.

In the view of machine learning, there are several methodologies for training. Some use Support Vector Machine for hand modeling. It needs appropriate amounts of dataset of hand images. Some use *Hidden Markov Model* (HMM) for dynamic hand gesture. Using stereo camera, model trains the specific 3D points of hands.

We focus on the static gesture recognition. Our approach is as follow:

- Using depth data from kinect, we can easily get the segmented hand images.
- To reduce the effect of lighting condition, we use the HOG feature extraction as a estimation parameter.
- From the dataset, we can generate the multi-class SVM classification model.
- After that, we implement these and put in the real-time Graphic User Interface framework, so we can check the depth data frame by frame.

3 Background

In this section, we will explain the background knowledge.

3.1 American Sign Language

`American Sign Language` (ASL) is predominant language for hearing-loss people. ASL possesses a set of 26

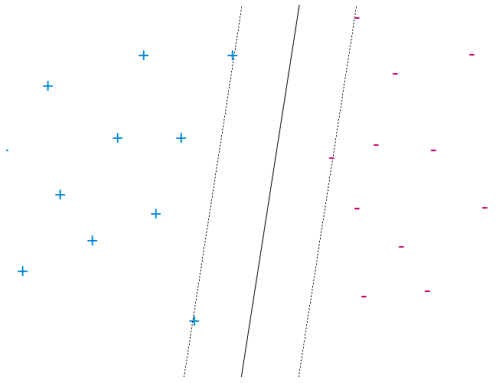


Figure 3: The optimal hyperplane

signs known as the American manual alphabet. It can be used to spell out words from the English language. These signs make use of the 19 hand shapes of ASL. That is, for example, the alphabet k and p are same hand shape but different meaning because of the orientation. Figure 2 shows the example of ASL, fingerspelling. Since this has identical characteristics, ASL could be the good example of hand gesture recognition.

3.2 Histogram of Oriented Gredient

`Histogram of Oritented Gredient` (HOG) are feature descriptors. This counts occurrences of the gradient orientation in localized portions of an image. This portion is called a *cell*. For improved accuracy, the local histograms can be contrast-normalized by exploring a measure of the intensity across a large area in images, called a *block*. It applies all of cells using this value to normalize. This normalized process help make better invariance to changes in illumination or shadowing.

The HOG descriptor has a few key advantages against to other descriptor-based methods. Since the HOG descriptor works on localized cells, the method supports invariance to geometric and photometric transformations, except for object orientation. Additionally, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization allow the individual body movement of objects to be ignored, if they maintain a roughly upright position. For these reasons, The HOG descriptor is particularly suited for human detection in images.

3.3 Support Vector Machine

`Support Vector Machine` (SVM) was developed by Vapnik and used to supervised learning. Basically, this machine is classifier of two sets which can be separable. It uses the support vector and kernels for learning. The kernel machine gives a framework which is flexible to the different domain by selecting the appropriate kernel functions. Unlike other machines, SVM makes a hyper plane or set of hyper planes in a high-dimensional space, which can be used for classification, regression, or other tasks. A

good separation would be achieved by the largest distance to the nearest training data point of any class hyper plain, since the classification contained large margin can get the lower error and higher generalization.

Figure 3 shows the optimal hyper plane. The SVM (primal) optimization problem is as follows:

minimize $\frac{1}{2}||w||^2$

subject to $y_t(w^T x_t + b) \geq 1, \forall t \in [1, N]$

This problem is quite complex to solve directly, we can compute this problem by formulating unconstrained optimization using Lagrange multipliers.

$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{t=1}^{N} \alpha_t [y_t(w^T x_t + b) - 1]$

where $\alpha_t \geq 0$

In this formula, we apply Karush-Kuhn-Tucker conditions, and then we arrive the dual fomula as follows:

maximize $\sum_{t=1}^{N} \alpha_t - \frac{1}{2} \sum_{s=1}^{N} \sum_{t=1}^{N} \alpha_s \alpha_t y_s y_t x_s^T x_t$

subject to $\sum_{t=0}^{N} \alpha_t y_t = 0$

where $\alpha_t \geq 0, \forall t \in [1, N]$

This dual problem has more simpler setting of involved constraints.

Although the original problem would imply in a finite dimensional space, it only occurs when the discriminated sets are not linearly separable in that space. Because of this, it was suggested that the original finite-dimensional space should be converted into a higher-dimensional space, for making the classification easily. To keep this process reasonably, the SVM is designed to ensure that data may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function.

4 Methodology

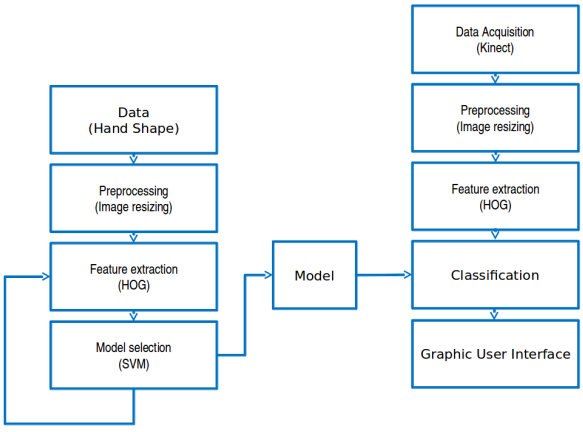


Figure 4: Procedure of the hand gesture recognition

The data which are images of hand gestures were taken by using Kinect, and we used HOG and SVM for image extraction and classification. The big picture of our procedure is in Figure 4.

4.1 Data

Figure 5 is training data. It is sorted by left to right, top to bottom. It contains alphabets A to Y except J because J and Z are dynamic hand gestures. Our goal is to recognize the static motions so that J and Z are excluded.

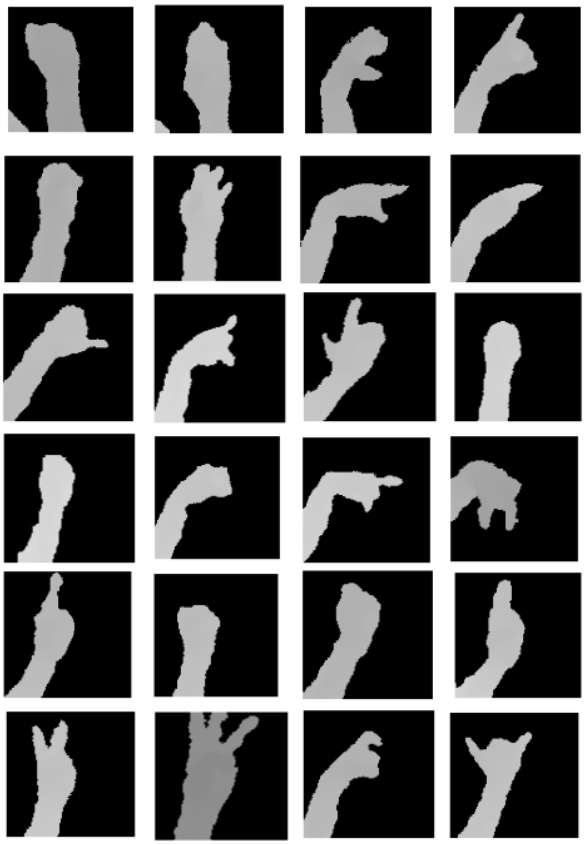
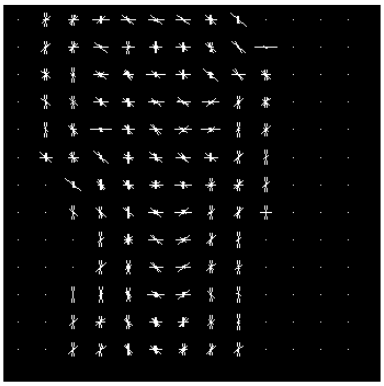


Figure 5: Training data, from A to Y except J, order by left to right, top to bottom

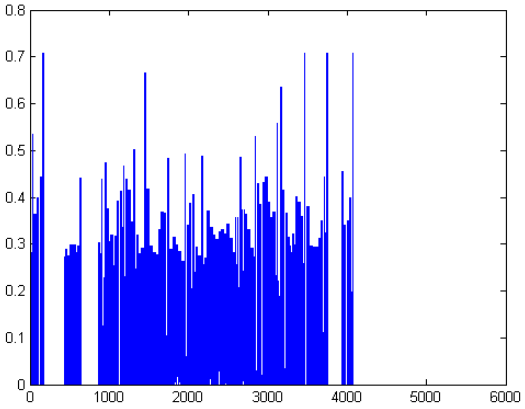
Matlab offers the Image Aqusition Toolbox for image processing. By attaching the kinect device, we can handle the color/depth frame in real time. Image Aqusition Toolbox gathers the 30 frames per sec from kinect. Each frame contains the *depthMetaData*. This skeleton data is one of the useful data to get hand position. From the data, we can get the joint position of right hand. We set the `Region Of Interest` (ROI) based on the center of mass. And we apply the background subtraction. From this process, *handDepthImage* is generated.

4.2 Image Extraction

The images are extracted and converted by HOG. We used HOG which is supported by MATLAB. The HOG returns the count of occurrences of oriented gradients.



(a) The gradients of an image



(b) The occurrence histogram of an image

Figure 6: Image extraction using HOG

The cells of HOG are described in Figure 6a. This represents the image oriented gradients. All of hand gestures are different in terms of this. Figure 6b shows that the occurrence of oriented gradients. We assume that each image would have unique histogram because an image has identical oriented gradients. By using that, we expected that those hand gestures can be classified.

4.3 Image Classification

We used SVM which is supervised learning models with associated algorithms that analyze data and recognize patterns. Since the number of gestures is 24, we need a multi-classifier and proper utility. LIBSVM [8] which supports multi-class classification is used. Using LIBSVM and MATLAB, we can classify the images.

5 Experimental Results

All experiments were executed on a PC with CPU (Intel Core i3-3220) 3.30GHz, 4GB RAM, and 500GB hard disk. The machine was operated by Windows 7. The

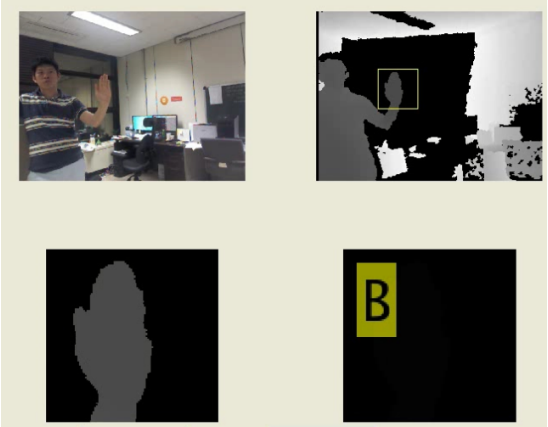


Figure 7: Graphical User Interface

code was implemented in MATLAB R2014a. Because we needed multi-classification SVM, we used LIBSVM [8]. 2,400 input hand gesture images were used for SVM training. That is, the training data is 100 images per alphabet. The test was executed by real-time. The Kinect gets 30 frames per second so that we count the correct recognition frame.

In the test, we made the graphical user interface (GUI) which is composed of four windows. The two windows on the top are taken through kinect. The left one is an original image and right one is a depth frame. From this depth data, the human body and hands are distinguished. The hand image is shown in left-bottom window. The last window shows the alphabet which is represented by a hand gesture. When the kinect detects the human, the framework generates the 18 skeleton points from depth data. Using right hand skeleton point, only right hand image is segmented, and compared it with trained model. If the decision is right, the right bottom window change the appropriate alphabet. For the test case, right decision of B is shown in figure 7.

Figure 8 shows the ambiguous gestures. All hand shapes are fist and have similar orientations. To solve this problem of recognition, we used depth data. Since the kinect provide the depth data, we used this information to classify the hand gestures.

Figure 9 is our results of simulation on the on-line test. In the on-line test, the hand gestures occurred randomly to clarify. For computing the match rate, we divided the matching frame count by total frame number. All hand gestures are classified well, except ambiguous gestures. The accuracy of all gestures is about 92.8% but that of ambiguous gestures is 72.9%.

6 Discussion

During the experiment, we found two considerations. First, kinect can obtain the depth frame data with 640 × 480 resolution. It can also gather the distance in pixels

Figure 8: Ambiguous gestures, A, E, M, N, S and T

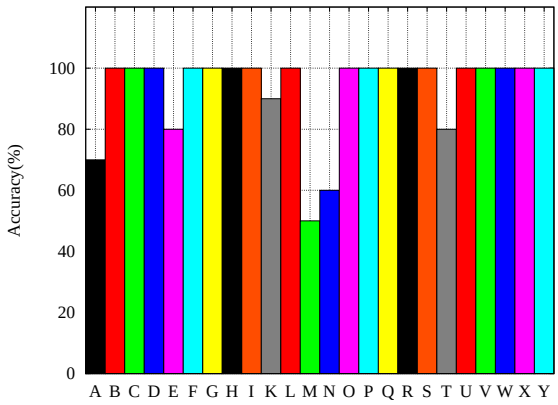


Figure 9: The accuracy of hand gestures

from data, but there are some limitations in software. Our base toolbox, MATLAB, does not support near-mode detection, which enabled to detect the distance at least 40cm from the device. Without this mode, it can only measure the distance at 80cm far from the kinect. Using the prototype implemented in C language, we can detect the finger line from grabbed hand. In contrast, our framework in MATLAB cannot separate the fingers in same pose. Thus, if we consider the feature selection in terms of depth resolution, we would get better recognition from ambiguous gestures.

Second, we just consider the gesture in static pose. There are some dynamic gestures such as J and Z in our case. Since HOG extracts the gradient of hand position rather than the sequence of that, it cannot train the dynamic gestures. In this situation, we regard the implementation of HMM approach. It can train the position of fingertip in fixed period. Using this position feature in SVM, we can train more challengeable dynamic gesture.

7 Conclusion

The HOG which is used feature extraction and SVM for classification of hand gestures were executed and applied in the real-time gesture recognition. Our experiments are motivated by specific situation for hearing-loss people. In our experiment, the results show that the recognition ratio is about 92.8%. If we have more dataset and better resolution to take the feature, the rate of recognition would be higher than that of our experiment. Some of alphabets are misclassified due to the gradient of hand gestures. By using the depth data from kinect, the ambiguous features will be recognized.

References

[1] Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2):52–73, 2007.

[2] Rung-Huei Liang and Ming Ouhyoung. A real-time continuous gesture recognition system for sign language. In *FG*, pages 558–567. IEEE Computer Society, 1998.

[3] Ying Wu and Thomas S. Huang. Vision-based gesture recognition: A review. In Annelies Braffort, Rachid Gherbi, Sylvie Gibet, James Richardson, and Daniel Teil, editors, *Gesture Workshop*, volume 1739 of *Lecture Notes in Computer Science*, pages 103–115. Springer, 1999.

[4] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *FGR*, pages 423–428. IEEE Computer Society, 2002.

[5] A. Kurakin, Z. Zhang, and Z. Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *EUSIPCO*, pages 1975–1979. IEEE, 2012.

[6] Xia Liu and Kikuo Fujimura. Hand gesture recognition using depth data. In *FGR*, pages 529–534. IEEE Computer Society, 2004.

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893. IEEE Computer Society, 2005.

[8] LIBSVM. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.