Static Hand Gesture Recognition Using Principal Component Analysis Combined with Artificial Neural Network

Trong-Nguyen Nguyen and Huu-Hung Huynh
DATIC, Department of Computer Science, University of Science and Technology, Danang, Vietnam
Email: ntnguyen.dn@gmail.com, hhhung@dut.udn.vn

Jean Meunier
DIRO, University of Montreal, Montreal, Canada
Email: meunier@iro.umontreal.ca

***Abstract*—Sign language is the primary language used by the deaf community in order to convey information through gestures instead of words. In addition, this language is also used for human-computer interaction. In this paper, we propose an approach which can recognize sign language, based on principal component analysis and artificial neural network. Our approach begins by detecting the hand, pre-processing, determining eigenspace to extract features and using artificial neural network for training and testing. This method has low computational cost and can be applied in real-time. The proposed approach has been tested with high accuracy and is promising.**

***Index Terms*—sign language, gesture, skin color, PCA, eigenspace, eigenvalue, eigenvector**

I. Introduction

Sign language is one of the several communication options used by people who are deaf or hard-of-hearing. This language uses signs made by moving the hands combined with facial expressions and postures of the body. The area of gesture language identification is being explored to help the community integration of the deaf and has high applicability. Most of the researchers use specialized equipment such as gloves or recognition techniques based on image processing through cameras and computers. Image processing solutions are usually based on two main methods: rules and machine learning. In this paper, we propose a new approach in the field of machine learning that can generalize hand gestures, and can be applied beyond the limit of usual hand gesture identification in the future using artificial neural network. The biggest advantage of our approach is the image resizing. Most researchers used the standard resizing methods, so the hand can be shrunk or stretched with a different (horizontal and vertical) ratio, and the obtained characteristics are affected. Our proposed resizing method can overcome this problem to improve recognition accuracy.

II. Related Work

Recently, some methods on gesture language recognition using cameras and image processing techniques have been implemented. The overall objective of these methods is to help disabled people communicate with each other, and replace traditional language by gesture language. Another type of gesture language applications is human–computer interaction, which uses gestures as input data, the information is transmitted to the computer via a webcam. Fujisawa [1] developed a human interface device (HID) to replace the mouse for the disabled. Bretzner [2] developed a system where users can control TV and DVD player based on hand gestures through a camera. Malima [3] proposed an algorithm that automatically identifies a limited set of hand gestures from images used for robot control to perform tasks. The largest disadvantage of these approaches is their high computational cost. Marshall [4] designed a system to support user interaction with multimedia systems, for drawing by gestures using a glove. However, this approach also used by other researchers is inconvenient for our purpose since the user must wear a special glove.

In gesture recognition, choosing features is a very important step because the hand gestures are diverse in shape, motion, variation and texture. Most of the features used in previous research subjects were extracted from the three following methods.

A. Hand Modeling (Model-Based Approach)

This approach tries to infer the pose of the palm and joint angles, it is ideal for interaction in virtual reality environments. A typical model-based approach may create a 3D model of a hand by using some kinematic parameters and projecting its edges onto a 2D space. Estimating the hand pose which in this case is reduced to the estimation of the kinematic parameters of the model is accomplished by a search in the parameters space for the best match between projected edges and the edges acquired from the input image. Ueda [5] estimated all joint angles to manipulate an object in the virtual space,

Manuscript received November 25, 2013; revised February 10, 2014.

doi: 10.12720/joace.3.1.40-45

the hand regions are extracted from multiple images obtained by the multi-viewpoint camera system. A hand pose is reconstructed as a “voxel model” by integrating these multi-viewpoint silhouette images, and then all joint angles are estimated using three dimensional matching between hand model and voxel model. Utsumi [6] used multi-viewpoint images to control objects in the virtual world. Eight kinds of commands are recognized based on the shape and movement of the hands. Bettio [7] presented a practical approach for developing interactive environments that allow humans to interact with large complex 3D models without having them to manually operate input devices. In model-based approaches, the initial parameters have to be close to the solution at each frame and noise is a real problem for the fitting process. Another problem is that it requires more time to design the system.

B. *View-Based Approaches*

These approaches model the hand by a collection of 2D intensity images. At the same time, gestures are modeled as a sequence of views. Eigenspace approaches are used within the view-based approaches. They provide an efficient representation of a large set of high dimensional points using a small set of orthogonal basis vectors. These basis vectors span a subspace of the training set called the eigenspace and a linear combination of these images can be used to approximately reconstruct any of the training images. The approach presented in [8] used this method. When using the appearance-based features, they achieved an error rate of 7%. Although these approaches may be sufficient for a small set of gestures, with a large gesture space collecting adequate training sets may be problematic. Another problem is the loss of compactness in the subspace required for efficient processing.

C. *Low-Level Features*

Some researchers presented a new and relatively simple feature space assuming that detailed information about the hand shape is not necessary for humans to interpret sign language. They found that all human hands have approximately the same hue and saturation, and vary primarily in their brightness. Using this color cue for hand segmentation, they used the low-level features of hand’s x and y position, angle of axis of least inertia, and eccentricity of the bounding ellipse. Some research used this method, such as [3]. Since the localization of hands in arbitrary scenes is difficult, one of the major difficulties associated with low-level features is that the hand has to be localized before extracting features.

III. Proposed Approach

In this section, we propose the needed process to recognize hand gestures. The letters of American Sign Language are shown in the Fig. 1.

A. *Input Data and Training Data*

In this approach, data can be an image or a sequence of images (video), taken by a single camera pointed toward the human hand. Some systems need two or more cameras to get more information about the hand pose. The advantage of these systems is that the gesture can be recognized even if the hand is occluded in one camera because the other cameras will capture the scene from different angles. However, the computational cost is also an issue.

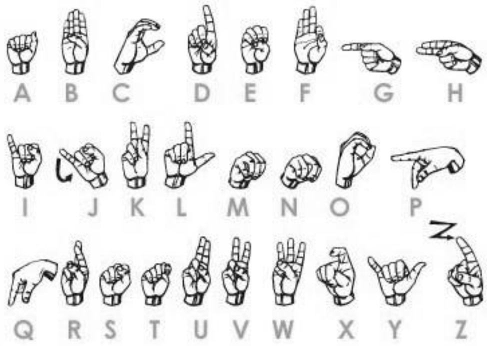

Figure 1. The letters of American sign language

In general, the accuracy of next stages in the identification process will be increased if the hand is detected easily. So the images are usually taken with a simple and homogeneous background environment which has high contrast with the skin color, and the shadow is limited in the obtained image.

The data used in this study were collected from open data sources and also captured in our laboratory.

Image: Images were collected from some of the open datasets [9], [10]. Some collected pictures are shown in Fig. 2.

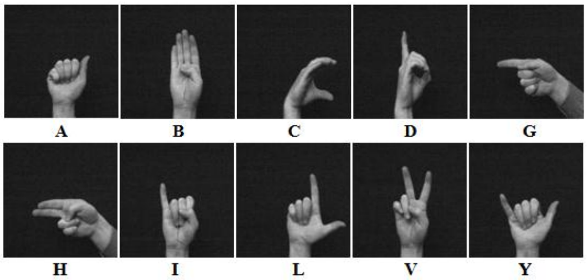

Figure 2. Some images from the dataset [9]

Video: Videos were recorded from a fixed webcam, with simple background and stable light. A person performed some gestures in front of the webcam. For easier segmentation, we did not show the face to the webcam. Videos were recorded by five different persons; each person performed a set of gestures, then they were transferred to AVI format (Audio Video Interleave) and tested.

B. *Pre-Processing*

These are necessary steps to segment the hand from the original frame.

Skin segmentation: To recognize the hand gesture, the first needed step is detecting the hand from the input picture. Two commonly used techniques are background subtraction and skin color filter. In the proposed solution, we use the second method.

Proposed by Fleck and Forsyth in [11], human skin color is composed by two poles of color: red (blood) and yellow (melanin), with medium saturation. Fleck also found that skin color has low texture amplitude. The skin color characteristics are essential information and can be used in hand tracking algorithm. Their skin color filter is proposed as follows: each pixel (RGB) is converted into log-component values *I*, R_g, and B_y using the following formulas:

$$L(x) = 106 * \log_{10}(x + 1 + n) \tag{1}$$

$$I = L(G) \tag{2}$$

$$R_g = L(R) - L(G) \tag{3}$$

$$B_y = L(B) - \frac{L(G) + L(R)}{2} \tag{4}$$

where *I*, R_g and B_y are respectively log-components with color channels Green, Red (minus green), Blue (minus green and red). The green channel is used to represent intensity because the red and blue channels from some cameras have poor spatial resolution. The constant 106 simply scales the output of the log function into the range [0,255], *n* is a random noise value, generated from a uniform distribution over the range [0, 1). The random noise is added to prevent banding artifacts in dark areas of the image. The constant 1 added before the log transformation prevents excessive inflation of color distinctions in very dark regions.

The log transformation makes the R_g and B_y values, as well as differences between *I* values (e.g. texture amplitude), independent of illumination level. Hue color at each pixel is determined based on *arctan* (R_g, B_y):

$$H = \frac{180}{\pi} \tan^{-1}(R_g, B_y) \tag{5}$$

Because the equation ignores intensity, so the result cannot distinguish the yellow and brown zones, and both will be considered the same. Saturation at each pixel is:

$$S = \sqrt{R_g^2 + B_y^2} \tag{6}$$

To compute texture amplitude, the intensity image is smoothed with a median filter, and the result subtracted from the original image. The absolute values of these differences are run through a second median filter.

If a pixel falls into either of the following ranges (see Fig. 3), it's a potential skin pixel:

texture < 5, 110 ≤ Hue ≤ 150, 20 ≤ Saturation ≤ 60
texture < 5, 130 ≤ Hue ≤ 170, 30 ≤ Saturation ≤ 130

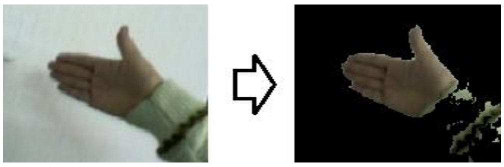

Figure 3. Skin color filter result

Keep the largest object: This step will retain a single object on the image. For the hand gesture identification system, the largest object appearing on the filtered image is the hand (see Fig. 4). So only the largest object is kept, others are removed. The detection accuracy can be improved by combining with some characteristics such as local binary pattern (LBP) [12], histogram of oriented gradients (HOG) [13].

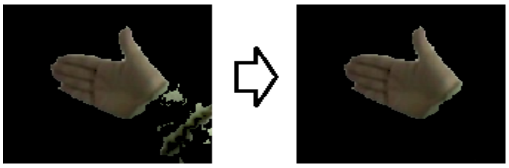

Figure 4. Keep the largest object and remove others

Applying median filter for gray image: In signal and image processing, it is often desirable to be able to perform some kind of noise reduction on an image or signal. The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise. The median filter result (size 3×3) is presented in the Fig. 5.

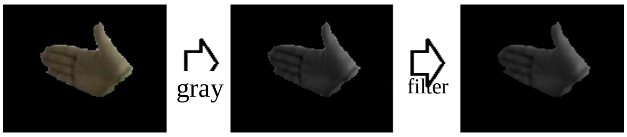

Figure 5. Median filter result

Image normalization: This step improves the balance and contrast of the image. Therefore, the recognition accuracy is almost unaffected when the image is obtained in many environments with different brightness. An example of this process is shown in Fig. 6.

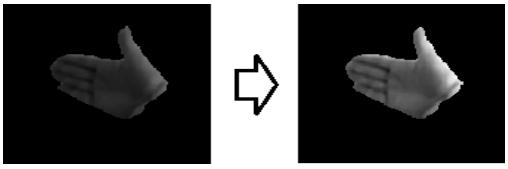

Figure 6. Normalized image

In some cases, the obtained image contains the arm. Before image normalization step, we remove the parts not related to the hand; this is an important step in the recognition process. When these components are removed, the near-away problem of the camera is eliminated. This not only affects the accuracy but also affects the processing speed - an important factor in real-time applications.

First, the hand is cropped by the object's bounding box. After that, we determine the wrist position and cut to separate the hand and arm. The wrist detection method is proposed as follows (see Fig. 7).

- Step 1: m_i is defined as object's width at row *i*

$$m_i = \sum pixel_{object} \in row_i$$

- Step 2: calculate *m* for the last row
- Step 3: calculate new *m* value for the line above

- Step 4: if *m* does not increase, go to step 3

else, crop image at the previous line

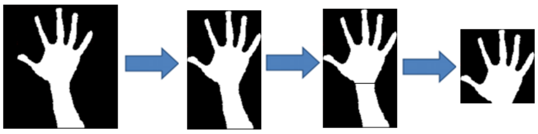

Figure 7. Locate the wrist and separate

Resizing Image: This step will standardize the hand image size, prepare for training and recognition, and increase the recognition accuracy. If we use the standard resizing methods, the hand can be shrunk or stretched with a different (horizontal and vertical) ratio, so the hand characteristics and the recognition results are affected. The hand must be cropped by the object's bounding box. Suppose that we have a hand image with size $w \times h$, where h and w are the height and width of the image, respectively. α is defined as the difference between w and h. So we propose a method to adjust the hand size as follows:

- If $h > w$ then $\alpha = h - w$
  - Insert $\alpha/2$ column(s) to the left of the image
  - Insert $\alpha/2$ column(s) to the right of the image
- If $h < w$ then $\alpha = w - h$
  - Insert $\alpha/2$ row(s) above the image
  - Insert $\alpha/2$ row(s) below the image
- Resize the obtained image to 100×100 pixels using standard resizing methods.

An example is shown in Fig. 8.

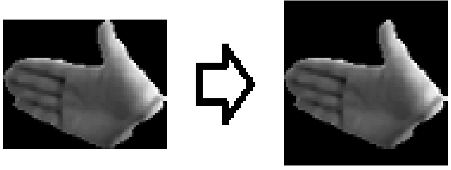

Figure 8. Resize the image with $h < w$

C. Recognition

Feature extraction: To extract the features from training images, we use the principal component analysis (PCA) algorithm [14]. This methodology is applied to our approach as below:

Step 1: obtain a set S with M training images; each image is transformed into a vector of size N = 10000 (100×100) and placed into the set.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3 \ldots \Gamma_M\} \tag{7}$$

Step 2: compute the mean image

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \tag{8}$$

Step 3: find the difference between each training image and the mean image

$$\Phi_i = \Gamma_i - \Psi \tag{9}$$

Step 4: obtain the covariance matrix C in the following manner

$$C = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T = \frac{1}{M}AA^T \tag{10}$$

where A = $[\Phi_1, \Phi_2, \Phi_3 \ldots \Phi_M]$

Step 5: we seek a set of M orthonormal vectors u_i which best describes the distribution of the data. The size of the matrix C is too big (10000×10000). So to find u_i, we seek eigenvectors v_i of the matrix $L = A^TA$ with size $M \times M$.

Step 6: compute eigenvectors of the matrix C based on eigenvectors of the matrix L

$$u_i = Av_i \tag{11}$$

Step 7: each training image Γ is transformed into new space and represented as a vector Ω

$$\omega = u_i^T(\Gamma - \Psi) \tag{12}$$

with $\Omega^T = [\ \omega_1, \omega_2, \omega_3, \ldots, \omega_M\ ]$

The obtained vectors are the feature vectors used for training. To recognize an image, the input is transformed into new space using the formula (12) and the obtained feature vector is put into the neural network.

Network training and recognition: Pattern recognition can be implemented by using a trained feed forward neural network. Multilayer neural network is a widespread disseminated effective method. In the proposed approach, the multilayer feed forward neural network is trained by the back propagation algorithm. Because of its easy implementation, fast and efficient operation, multilayer feed forward network is widely used in numerous applications. When the network is trained, the input patterns are recognized and distinguished by the associated output patterns. The network progressively adjusts the outputs with respect to input patterns until approaching an error criterion. Subsequently, the best network architecture is selected with a test dataset. After testing, the final network delivers the output corresponding to the input pattern with minimal error.

We designed a Multiple Layer Perceptron (MLP) network that has three layers: input layer has the number of neurons corresponding to the size of the feature vector (amount of training images); the number of neurons in the hidden layer (20) were determined by trial–and–error method; and the number of neurons in the output layer is the number of gestures with we need to recognize.

An activation function for a back-propagation network has several important characteristics, such as, continuous, differentiable, and monotonically non-decreasing. The desirable activation function used in this paper is the binary sigmoid function, which has a range of (0, 1) and is defined as:

$$tansig(n) = \frac{2}{1+e^{-2n}} - 1$$

IV. Experimental Results

We used a Logitech 9000 webcam for this research. In experiments, the distance from the webcam to the hand is ranging from 0.8 to 1.2m. Our system is implemented in C# language using the OpenCVSharp library. We identified 24 letters of the alphabet (without J and Z), with 150 training gestures collected from datasets [9], [10] and our laboratory for each letter. The letters J and Z are not used for this research because of their non-static motion.

The gestures were tested by performing them directly in front of the webcam. Each gesture is recognized with 100 images, corresponding to a set of 2400 gestures. The average positive recognition rate reached 94.3%, with 3600 training images. Testing results are shown in Fig. 9.

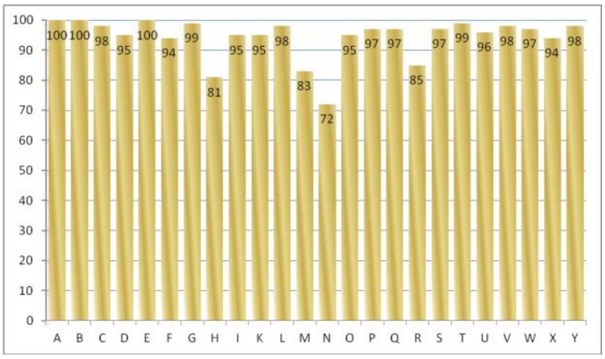

Figure 9. Positive recognition rates with 24 letters

Some letters were recognized perfectly: A, B, E or with a high rate: C, D, F, G, I, K, L, O, P, Q, S, T, U, V, W, X, Y (over 94%). However, there are still some gestures that have been less successful in recognizing: H, M, N and R. One reason is the high similarity between images that represent these characters. For instance, the silhouettes of M and N hand gestures are nearly the same. Other reasons are the large hand inclination and that the hand is not directly opposite to the camera.

V. Conclusion and Discussion

In this paper, a new approach was proposed to recognize sign language. The system consists of the following process: segmentation, pre-processing, features extraction, training and identification. In the detection step, color information (hue and saturation) is used to highlight the skin in the image. The pre-processing enhances image quality and gets the hand without arm. Then the characteristics of each hand are extracted based on PCA method, and an artificial neural network is used for recognition. The focus of this research is the resizing method which can classify different gestures, and applying PCA for feature extraction, with low computational cost features for identification. Furthermore, our system is easy to install and can execute in real-time.

The accuracy of our approach compared with other methods on the same samples is presented in the Table I.

TABLE I. Methods Comparison

Approach	Our	[15]	[16]	[17]	[18]
Recognition	94.3%	84%	92.78%	90.45%	91.6%

As further work, to increase the efficiency of recognition (especially H, M, N and R), some additional features based on the depth of the hand will be extracted via Kinect. In addition, we can extract the information about the hand's direction, so that the gestures can be recognized even when the hand inclination is large, or the hand is not directly opposite to the camera.

Acknowledgment

This work was supported by the DATIC, Department of Computer Science, University of Science and Technology (DUT), The University of Danang, Vietnam and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] S. Fujisawa, *et al.*, "Fundamental research on human interface devices for physically handicapped persons," in *Proc. 23rd Int. Conf. IECON*, New Orleans, 1997.

[2] S. Lenman, L. Bretzner, and B. Thuresson, "Computer vision based hand gesture interfaces for human-computer interaction," *Department of Numerical Analysis and Computer Science*, June 2002.

[3] A. Malima, E. Ozgur, and M. Cetin, "A fast algorithm for vision-based hand gesture recognition for robot control," in *Proc. IEEE Conference on Signal Processing and Communications*, 2006, pp. 1-4.

[4] M. Marshall, "Virtual sculpture-gesture controlled system for artistic expression," in *Proc. AISB COST287-ConGAS Symposium on Gesture, Interfaces for Multimedia Systems*, Leeds, UK, 2004, pp. 58-63.

[5] E. Ueda, "A hand pose estimation for vision-based human interfaces," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 676-684, 2003.

[6] A. Utsumi and J. Ohya, "Multiple hand gesture tracking using multiple cameras," in *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, 1999, pp. 473-478.

[7] F. Bettio, *et al.*, "A practical vision-based approach to unencumbered direct spatial manipulation in virtual worlds," in *Proc. Eurographics Italian Chapter Conf*, 2007.

[8] N. Gupta, *et al.*, "Developing a gesture based inter-face," *IETE. Journal of Research: Special Issue on Visual Media Processing*, 2002.

[9] S. Marcel. Hand Posture and Gesture Datasets. [Online]. Available: www.idiap.ch/resource/gestures

[10] Thomas Moeslund's Gesture Recognition home page. [Online]. Available: www-prima.inrialpes.fr/FGnet/data/12-MoeslundGesture

[11] M. Fleck, D. Forsyth, and C. Bregler, "Finding naked people," in *Proc. of European Conference on Computer Vision*, 1996.

[12] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037-2041, Dec 2006.

[13] A. Mittal, A. Zisserman, and P. Torr, "Hand detection using multiple proposals," in *Proc. British Machine Vision Conference*, September 2011, pp. 75.1-75.11.

[14] L. I. Smith, *A Tutorial on Principal Component Analysis*, Cornell University, USA, 2002.

[15] M. M. Hasan and P. K. Mirsa, "Brightness factor matching for gesture recognition system using scaled normalization," *International Journal of Computer Science & Information Technology*, vol. 3, no. 2, 2011.

[16] V. S. Kulkarni and S. D. Lokhande, "Appearance based recognition of american sign language using gesture segmentation," *International Journal on Computer Science and Engineering*, vol. 2, no. 3, pp. 560-565, 2010.

[17] S. Y. Zhao, W. J. Tan, S. G. Wen, and Y. Y. Liu, "An improved algorithm of hand gesture recognition under intricate background," in *Proc. Springer the First International Conference on Intelligent Robotics and Applications*, Part I., 2008, pp. 786-794.

[18] E. Stergiopoulou and N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," *Elsevier Engineering Applications of Artificial Intelligence*, vol. 22, no. 8, pp. 1141-1158, 2009.

Trong-Nguyen Nguyen received the B.S. degree in information technology from the Danang University of Technology, Vietnam, in 2012. Now, he is a master student at his studied university. His research work focuses on computer vision and machine learning.

Huu-Hung Huynh received the B.S. degree in physics from the Hanoi University of Technology, Vietnam, in 1998, the M.Sc.A. degree in Computer Science in 2003, and the Ph.D. degree in Computer Science from the Aix-Marseille University in 2010. He now is lecturer at University of Science and Technology, Danang, Vietnam. His current research interests include computer vision and its applications to medical imaging and health care.

Jean Meunier received the B.S. degree in physics from the Université de Montréal, Canada, in 1981, the M.Sc.A. degree in applied mathematics in 1983, and the Ph.D. degree in biomedical engineering from the École Polytechnique de Montréal, Canada, in 1989. In 1989, after postdoctoral studies with the Montreal Heart Institute, he joined the Department of Computer Science and Operations Research, Université de Montréal, where he is currently a Full Professor. He is also a regular member of the Biomedical Engineering Institute at the same institution. His current research interests include computer vision and its applications to medical imaging and health care.