Static Hand Gesture Recognition Using Artificial Neural Network

Trong-Nguyen Nguyen, Huu-Hung Huynh
DATIC, Department of Computer Science, Danang University of 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—**Computers are widely used in all fields. However, the interaction between human and machine is done mainly through the traditional input devices like mouse, keyboard etc. To satisfy the requirements of users, computers need other ways to interact more conveniently, such as using speech or body language (e.g. gestures, posture). In this paper, we propose a new method supporting hand gesture recognition in the static form, using artificial neural network. The proposed solution has been tested with high accuracy (98%) and is promising.**

Index Terms—**gesture recognition, sign, boundary shape, cross section, skin color**

I. Introduction

Sign language is a language that employs signs made by moving the hands combined with facial expressions and postures of the body. It is one of several communication options used by people who are deaf or hard-of-hearing. Gesture language identification is one of the areas being explored to help the deaf integrate into the community and has high applicability. Researchers use specialized equipment such as gloves or recognition techniques based on image processing through cameras and computers. Most image processing solutions are based on two main methods: rules and machine learning. In this paper, we propose a new method 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 an artificial neural network and where the main contribution is in the feature extraction.

II. Related Work

Recently, some subjects on gesture language recognition using cameras and image processing techniques have been implemented. The overall objective of these subjects 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, that uses gestures as input data, the information is transmitted to the computer via a webcam. Bretzner [1] developed a system where users can control TV; DVD player based on hand gestures through a camera. Malima [2] proposed an algorithm that automatically identifies a limited set of hand gestures from images used for robot control to perform tasks. Fujisawa [3] developed a communication device HID to replace the mouse for the disabled. Marshall [4] designed a system to support user interaction with multimedia systems, drawing by gestures.

In hand gesture recognition, the selection of features is very important 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.

Hand modeling (model-based approach): this method tries to infer the pose of the palm and joint angles, is ideal for interaction in virtual reality environments. A typical model-based approach may create a 3D model of a hand by using some kinematics parameters and projecting its edges onto a 2D space. Estimating the hand pose which in this case is reduced to the estimation of the kinematics 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 [9] estimates all joint angles to manipulate an object in the virtual space, 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 [11] 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 [12] 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

Manuscript received December 10, 2012, revised February 1, 2013; accepted April 10, 2013.

©2013 Engineering and Technology Publishing
doi: 10.12720/joig.1.1.34-38

problem for the fitting process. Another problem is that it requires more time to design the system.

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. These approaches were used in many of the hand gesture recognition subjects, such as [13]. 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.

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 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 [2]. 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 method to recognize hand gestures. We use artificial neural network because of its power and flexibility.

The following will present the main steps in our method.

A. Input Data and Training Data

Data can be an image or a sequence of images (video), taken by a single camera toward the human hand. However, some systems use two or more cameras to get more information about the hand pose. The advantage is that the system can recognize the gesture even if the hand is obscured for one camera because the other cameras will capture the scene from different angles.

A different system was presented with the camera mounted on a hat to take the area in front of the wearer. The obvious advantage of this system is the camera position is always appropriate if people move or turn around.

In general, the following stages of the identification process will be less complex if the image is taken with a simple background and the contrast is high with the hand. So pictures are usually taken in a homogeneous background environment, and limit shadows in the obtained image.

The used data were collected from several open data sources.

Image: Images were collected from some of the open data sets [14] [15] and only a few pictures of symbols A, B, C, D, G, H, I, L, V, Y in the American Sign Language (ASL) table are used. Here are some collected pictures:

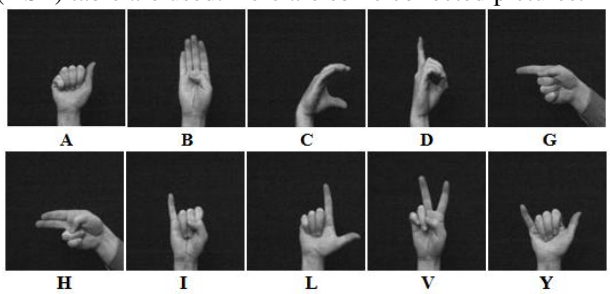

Figure 1. Some images from the data set [14]

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

B. Pre-Processing

These are the necessary steps to get the complete hand picture from the original frame.

Hand detection: To identify the hand gesture, the first needed step is detecting the hand from the input frame. 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 [16], 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 a low amplitude structure. The skin color characteristics are essential information and can be used in hand tracking algorithm. Skin Color Filter is proposed as follows: each pixel (RGB) is converted into log-component values I, R_g, and B_y [16] using the following formulas:

$$L(x) = 105 * \log_{10}(x + 1 + n) \quad (1)$$

$$I = L(G) \quad (2)$$

$$R_g = L(R) - L(G) \quad (3)$$

$$B_y = L(B) - (L(G) + L(R))/2 \quad (4)$$

Where I, R_g, B_y are respectively log-components with color channels Green, Red, Blue.

The green channel is used to represent intensity because the red and blue channels from some cameras have poor spatial resolution. The constant 105 simply scales the output of the log function into the range [0,254]. n is a random noise value, generated from a distribution uniform 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

©2013 Engineering and Technology Publishing

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):

$$Hue = 180/\pi \tan^{-1}(R_g, B_y) \quad (5)$$

Saturation at each pixel is $\sqrt{R_g^2 + B_y^2}$. Because the equation ignores intensity, so the result cannot distinguish the yellow and brown zones, and both will be considered yellow.

$$Saturation = \sqrt{R_g^2 + B_y^2} \quad (6)$$

When the color and saturation are calculated, the skin region can be marked by using the pixels which values have the following attributes:

$$110 \le Hue \le 150 \text{ and } 20 \le Saturation \le 60$$

$$130 \le Hue \le 170 \text{ and } 30 \le Saturation \le 130$$

Figure 2. Skin color filter result

Median filter: In signal 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 is presented below:

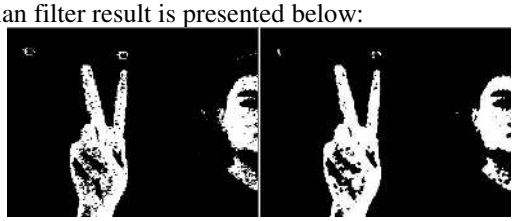

Figure 3. Median filter result

Select the largest object: This step helps to retain a single object on the image. For the identification system of hand gestures, the largest object appearing on the resulting image is the hand. So after this step, the remaining object is the hand needed to identify the gesture meaning.

Figure 4. Select the largest object

Fill holes inside the object: In the proposed method, we use some features that require the object must be filled to be a block. To do this, we use the flood fill algorithm.

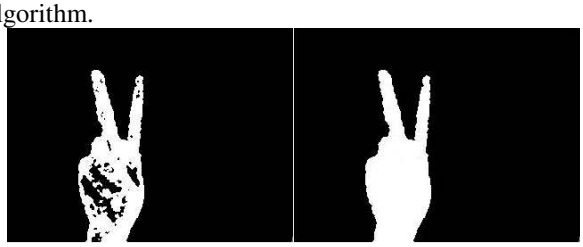

Figure 5. Fill the holes

Remove the arm: We skip the parts not related to the hand; this is an important step in the identification process. When we remove these components, the problem near - away of the camera is removed, and the resulting image is the hand. This not only affects the accuracy but also affects the processing speed - an important factor in real-time applications. First, the image is resized by the object bounding box size.

To get the region around the hand, we determine the position of the wrist and cut to separate the hand and arm. The wrist detection algorithm is proposed as follows.

- Step 1: m_i is defined as object's width at row i
  $m_i = pixel_{object} \times row_i$
- 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

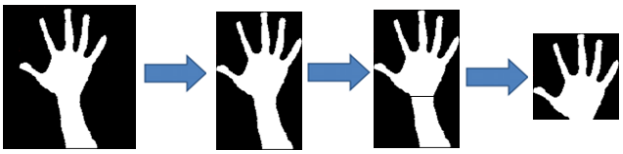

Figure 6. Locate the wrist and separate

C. Selected Features

This is the process of creating a set of attribute's descriptions for the image. The descriptions are stored as a feature vector for using in the training and identification process. The three main features are the change of the horizontal / vertical object pixels, the shape of boundary, and the scalar description.

The change of object pixels: This feature represents the changes of pixel values through cross sections, which are split evenly on the object. The number of cross sections depends on the level of detail that we want to extract. More cross sections mean that longer information is extracted from the object, but the complexity and storage capacity will be increased. The number of cross sections can be chosen in a flexible way to get the best number of features. The figure below describes the cross sections and each corresponding value by calculating the number of changes from the background pixel to the foreground pixel and the contrary.

©2013 Engineering and Technology Publishing

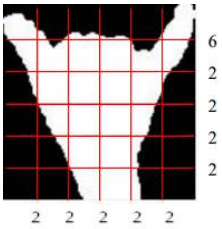

Figure 7. The change with 10 cross sections

The boundary shape: This feature calculates the distance from an outside edge to the hand edge in a particular direction. In this paper, we use the object shape characteristics from three edges: left, right and above (e.g. three histograms in the Fig. 8). In each graph, the horizontal axis represents the image border, and the vertical axis is the distance between the edge of the hand (set of pixels of the edge) and image border in one direction. These features are calculated by dividing the boundary into *n* sections, and we compute each segment average of the histogram, then we have *n* elements of the feature vector.

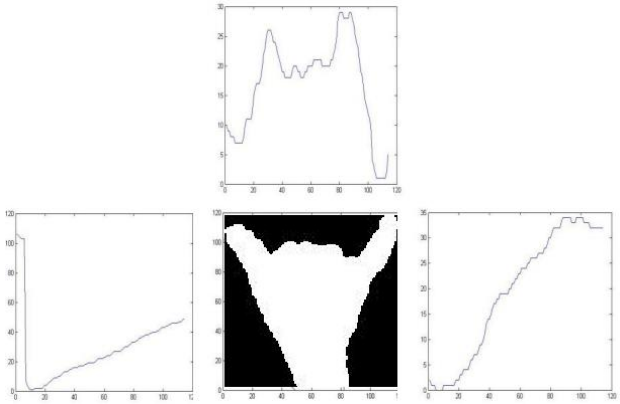

Figure 8. The boundary shape from three edges

The scalar description:

- Ratio between the width *w* and height *h* of the hand

$$Edges\ ratio = \frac{w}{h} \tag{7}$$

- The ratio of hand area and bounding box area

$$Area\ ratio = \frac{\sum pixel_{foreground}}{w * h} \tag{8}$$

Establish the feature vector: The feature vector is set by combining the three features described above. To identify the best features, we conducted experiments with different numbers of attributes of each feature.

TABLE I. THE FEATURE VECTOR COMPONENTS

Feature vector	Features					Vector size
	Object crossing	Boundary shape			Scalar description	
		Left	Above	Right		
FV1	4	6	6	6	2	24
FV2	6	8	8	8	2	32
FV3	8	10	10	10	2	40
FV4	10	12	12	12	2	48

D. Train Network and Recognition

We designed a Multiple Layer Perceptron network that has three layers: input layer has the number of neurons corresponding to the size of feature vector; the neurons of hidden layer determined by trial–and–error method; and output layer has 10 neurons matching 10 gestures.

The used transfer function is a hyperbolic tangent sigmoid (tansig):

$$tansig(n) = \frac{2}{1 + e^{-2n}} - 1 \tag{9}$$

IV. EXPERIMENTAL RESULTS

We used the Logitech 9000 webcam to serve this research. Our method gives good results in spite of the medium quality images of our acquisition system. In experiments, distance of the webcam to the hand is about 0.8 – 1.2m. Our system is implemented in Matlab and C++ using the OpenCV library.

We selected 10 letters of the alphabet to identify, including A, B, C, D, G, H, I, L, V, Y.

Training data consist of 450 samples taken from [14] [15], in which a gesture has many different viewing angles. Test data include 445 random samples collected from different sources.

Test results for the feature vector FV1, FV2, FV3 and FV4 is shown in the table below:

TABLE II. RECOGNITION RATES WITH DIFFERENT FEATURES

Feature vector	Number of elements	Positive rate
FV1	24	97.1%
FV2	32	97.3%
FV3	40	98.0%
FV4	48	98.0%

The best achieved results give an accuracy of up to 98% when using the characteristic vector FV3, FV4. This shows that the accuracy increases when the number of vector element increases, but if too many elements, the processing speed will decrease. Therefore, we need more test to select the characteristic components to ensure accuracy and processing speed. According to Table II, vector FV3 with 40 characteristic elements is selected for the next test.

For video data, we do the following steps to determine the frame that contains hand gesture to be identified:

- Determine the feature vector $v_t = (v_1, v_2, \ldots v_n)$ of the hand in the frame *t*
- Define V_t as the point with coordinates $(v_1, v_2, \ldots v_n)$ in *n*-dimensional space
- We consider *n* (e.g. 24) continuous frames from the current to the previous, if the Euclidean distance between V_t and V_{t-1} is smaller than a given constant α, the hand gesture in the current frame must be identified.

©2013 Engineering and Technology Publishing

TABLE III. TEST RESULTS WITH VIDEO DATA

Video	Frame size	Total frames	Real gesture	True recognize	Positive rate
1	320 x 240	1527	10	9	90%
2	320 x 240	1035	10	8	80%
3	320 x 240	1010	10	8	80%
4	320 x 240	920	10	9	90%
5	320 x 240	1100	10	9	90%

In some cases, recognition results are not correct because the hand inclination is large; the hand is not directly opposite to the camera and gesture similarity between the letters (see Fig. 9).

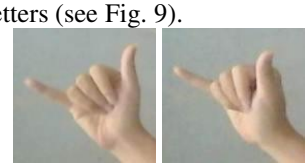

Figure 9. Some cases that the recognition results are not correct

V. CONCLUSION AND DISCUSSION

In this paper, a new solution was proposed to identify hand gesture. The system consists of the following process: hand detection, pre-processing, features extraction, network training and identification. The focus of this research is selecting the features which can classify different gestures, so the main advantage of this system is that it takes low computational cost features for identification, and our system is easy to install and can execute in real-time.

TABLE IV. METHODS COMPARISON

Method	Our	[5]	[6]	[7]	[8]	[10]
Recognition Percentage	98%	84%	92.78%	90.45%	98.3%	90.45%

However, some limitations still need to be overcome to make this method more effective, such as for differentiating significant gestures or supplement characteristics to distinguish some gestures often mistaken (‘G’ is sometimes recognized as ‘A’, ‘D’). Using additional features is also a subject to be explored in the future work.

ACKNOWLEDGMENTS

This work was supported by the DATIC, IT Faculty, Danang University of Technology, Vietnam and DIRO, Montreal, Canada.

REFERENCES

[1] 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.

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

[3] S. Fujisawa, et al, “Fundamental research on human interface devices for physically handicapped persons,” *23rd Int. Conf. IECON*, New Orleans, 1997.

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

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

[6] 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.

[7] S. Zhao, W. Tan, S. Wen, and Y. Liu, “An improved algorithm of hand gesture recognition under intricate background,” *Springer the First International Conference on Intelligent Robotics and Applications (ICIRA 2008)*, Part I. pp. 786–794, 2008.

[8] B-W Min, H-S Yoon, J. Soh, Y-M Yang, and T. Ejima, “Hand gesture recognition using hidden Markova models,” *IEEE International Conference on computational cybernetics and simulation*, vol. 5, 1997.

[9] E. Ueda, “A hand pose estimation for vision-based human interfaces,” *IEEE Transactions on Industrial Electronics*, vol. 50, No. 4, pp. 676–684, 2003.

[10] 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.

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

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

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

[14] [Online]. Available: http://www.idiap.ch/resource/gestures

[15] [Online]. Available:http://www-prima.inrialpes.fr/FGnet/data/12-MoeslundGesture

[16] M. Fleck, D. Forsyth, and C. Bregler, “Finding naked people,” *European Conference on Computer Vision*, 1996.

Trong-Nguyen Nguyen received the B.S. degree in IT Faculty from the Danang University of Technology (DUT), 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 Computer Science from the IPH, 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 DUT, Vietnam. His current research interests include computer vision and and health care systems.

Jean Meunier received the B.S. degree in physics from the University of Montreal (UdeM), Canada, in 1981, the M.Sc.A. degree in applied mathematics in 1983, and the Ph.D. degree in biomedical engineering from the Polytechnique of Montreal in 1989. In 1989, after postdoctoral studies with the Montreal Heart Institute, he joined the DIRO, UdeM, 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.

©2013 Engineering and Technology Publishing