ISSN No. 0976-5697

Volume 4, No. 11, Nov-Dec 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Vietnamese Sign Language Recognition using Cross Line Descriptors and Invariant Moments

Trong-Nguyen Nguyen
DATIC, Department of Computer Science
University of Science and Technology
Danang, Vietnam
ntnguyen.dn@gmail.com

Huu-Hung Huynh
DATIC, Department of Computer Science
University of Science and Technology
Danang, Vietnam
hhhung@dut.udn.vn

Jean Meunier
DIRO, University of Montreal, Montreal, Canada
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. Many researches about this language have been done, and they play an important role in life. Currently, most of the hard-of-hearings in Vietnam are facing many difficulties in community integration because of their limited ability. So we propose an approach which can recognize Vietnamese sign language, based on digital image processing combined with a machine learning method. After pre-processing, we use a combination of cross lines descriptors and invariant moments to extract the features, and then the gesture is recognized using support vector machines. This is one of the few studies on sign language applied to Vietnamese alphabet (the number of words is higher and more complex than international standards with several accented letters). The proposed approach has been tested with high accuracy and is promising.

Keywords: gesture; sign; skin color; cross line; invariant moment; kernel function; one-against-all; one-against-one.

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 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 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 support vector machines (SVMs).

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, that 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 [4] 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 [3] 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.

There is very few researches applied for the Vietnamese sign language. So it is necessary to have more studies to support communication between deaf people, as well as for the human–computer interaction in Vietnam. A solution [5] was proposed in 2005, which can recognize the hand gesture via a glove fitted with a sensors system. However, it is difficult to apply this method to the deaf community in Vietnam.

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. 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. Ueda [7] estimated 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. Bettio [8] 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 [9] 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 [4]. 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 steps for recognizing hand gestures. We use the SVM because of its power and flexibility. The widely used sign language in Vietnam is shown in the Fig. 1.

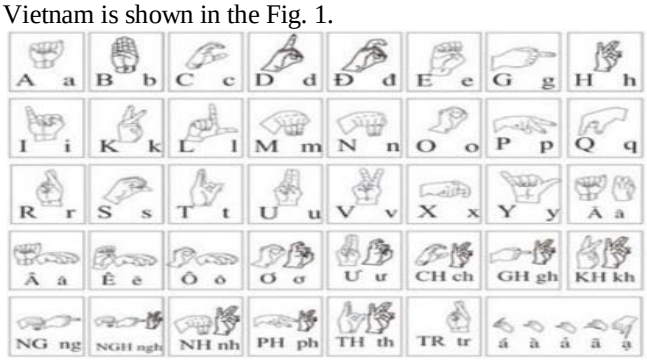

Figure: 1 Widely used Vietnamese sign language

We see that the special Vietnamese characters (ă, â, ê, ô, ơ, and ư) are represented by two hands, one is the letter of the international standard alphabet; the rest is a characteristic symbol. Therefore, Vietnamese characters can be recognized by combining two identification results of two hands.

The following will present the main steps in our approach.

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 most 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.

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.

a. ***Image:*** Images were collected from some of the open datasets [15, 16]. Some collected pictures are shown in Fig. 2.

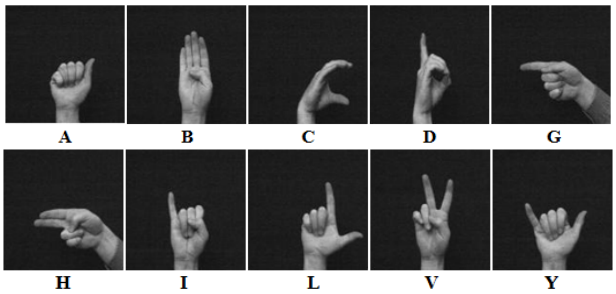

Figure: 2 Some images from the dataset [15]

b. ***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.

a. ***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 [17], 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:

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

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

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

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

Where I, R_g 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):

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

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 the same.

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

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:

$$\text{texture} < 5,\ 110 \le Hue \le 150,\ 20 \le Saturation \le 60$$

$$\text{texture} < 5,\ 130 \le Hue \le 170,\ 30 \le Saturation \le 130$$

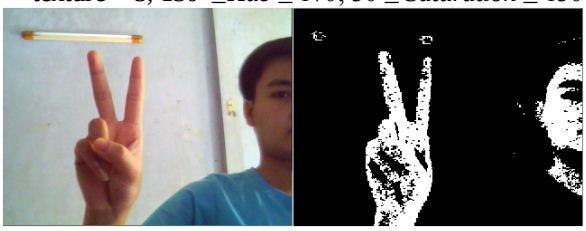

Figure:2 Skin Color Filter result

b. Median filter: 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. 4.

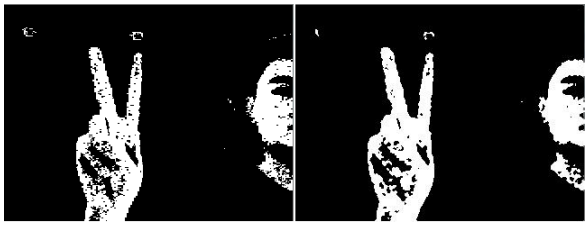

Figure: 3 Median filter result

c. 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. 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) [21], or histogram of oriented gradients (HOG) [22].

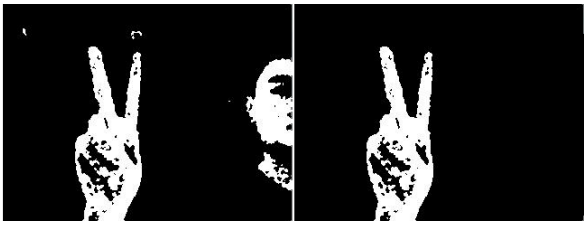

Figure: 4 Keep the largest object and remove others

d. Fill holes inside the object: To describe fully the shape of the hand, the holes inside the object are filled. To do this, we use the flood fill algorithm [19].

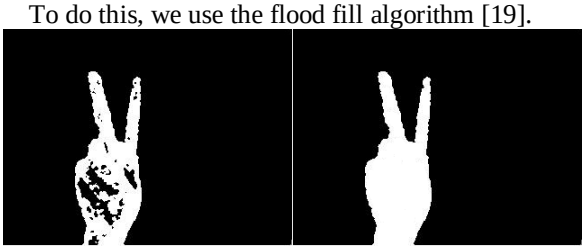

Figure:5 Fill the holes

e. Remove the arm: We skip 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).

(a). Step 1: m_i is defined as object's width at row i

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

(b). Step 2: calculate m for the last row
(c). Step 3: calculate new m value for the line above
(d). Step 4: if m does not increase, go to step 3
else, crop image at the previous line

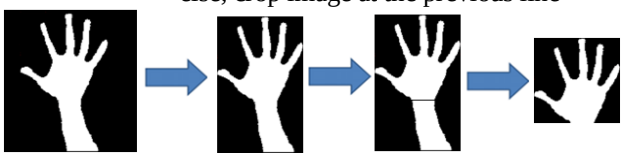

Figure:6 Locate the wrist and separate

C. *Features extraction and recognition:*

a. ***Resizing image:*** First, the hand is resized to be a 100×100 image. This step will standardize the hand image size, prepare for extracting feature vector, 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 characteristics and consequently the recognition results are affected. Suppose that we have a binary hand image with size $w \times h$, where h is the height and w is the width of the image. α is defined as the difference between w and h. So we propose a method to adjust the hand size as follows:

(a). If $h > w$ then $\alpha = h - w$
- Insert $\alpha/2$ column(s) to the left side
- Insert $\alpha/2$ column(s) to the right side

(b). If $h < w$ then $\alpha = w - h$
- Insert $\alpha/2$ row(s) above the image
- Insert $\alpha/2$ row(s) below the image

(c). Resize the obtained image to 98×98 pixels using standard resizing methods.

(d). Insert one column to the left of left side and one to the right of right side. Similarly, we insert one row above and one below the image. The obtained image has size 100×100.

Some examples are shown in Fig. 8 and 9.

Figure:7 Resize the image with h < w

Figure: 8 Resize the image with h > w

b. ***Extracting cross line descriptor:*** This feature represents the changes of pixel values through cross lines, which are split evenly on the object. The number of cross lines depends on the level of detail that we want to extract. More cross lines mean that longer information is extracted from the object, but the complexity and storage capacity will be increased. The number of cross lines can be chosen in a flexible way to get the best number of features. The Fig. 10 describes the cross lines and each corresponding value by calculating the number of changes from the background pixel to the foreground pixel and the contrary. In this approach, we use 10 horizontal and 10 vertical cross lines for describing object. Therefore, we have 20 elements of feature vector.

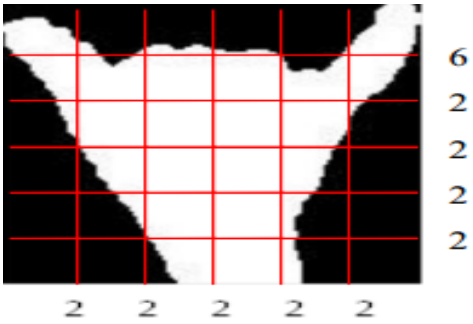

Figure:9 The change with 10 cross lines

c. ***Calculating invariant moments:*** To determine invariant features in scaling and rotating, we use the normalized central moments defined as (Hu, 1962):

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{7}$$

Where $\gamma = \frac{p+q}{2} + 1 \quad \forall p+q \geq 2$, μ_{pq} values are centralized moments that are described in [23].

There are seven invariant moments, and we use five first values for recognizing, consist two second-order and three third-order moments:

$$M1 = \eta_{20} + \eta_{02} \tag{8}$$

$$M2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{9}$$

$$M3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{10}$$

$$M4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{11}$$

$$M5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{03} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \tag{12}$$

By combining these 5 invariants and 20 values described earlier, we have the feature vector that contains 25 elements for recognizing.

d. ***Training and recognition:*** In machine learning, SVMs are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In more details, when the data cannot be separated by a hyper plane in their original domain, we can project them into a higher dimensional Hilbert space [20] and attempt to linearly separate them in the new space using kernel functions. Therefore, the decision boundary is given by

$$f(x) = sign(\sum a_i y_i K(x, x_i) + b) \tag{13}$$

Where $K(x, x_i)$ is a kernel function, a_i and b are parameters and y_i represents one of the two classes (y_i = 1 or -1). Frequently used kernel functions are the linear kernel, the polynomial kernel, and the Gaussian radial basis function kernel, which is defined as

$$K(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2) \qquad (14)$$

Since sign language recognition is applied for more than two gestures, we used the multi-class SVM for classification. One common method is building binary classifiers which distinguish between one of the labels and the rest (one-against-all) or between every pair of classes (one-against-one). We used the one-against-one approach because of the large number of classes [18]. The classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one, and finally the class with the most votes determines the instance classification.

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 selected 23 letters of the alphabet to identify, with 108 training images collected from datasets [15, 16] and our laboratory for each letter.

After some training processes with different kernel functions like RBF, linear, polynomial with different parameters, we found that the linear kernel function gives the highest accuracy, and uses fewer parameters.

The gestures were then tested by performing them directly in front of the webcam. Each gesture is recognized 100 samples, corresponding to a set of 2300 gestures. The average positive recognition rate reached 95%, with 2484 training images. The testing results are shown in the Fig. 11.

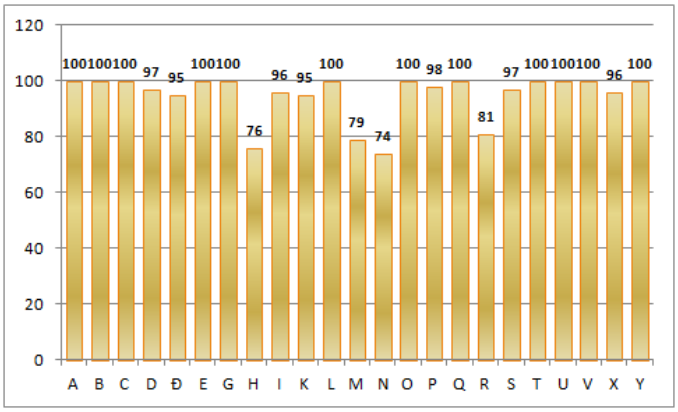

Figure: 10 Positive recognition rates with 23 letters

Many letters were recognized perfectly: A, B, C, E, G, L, O, Q, T, U, V, Y, or with a high rate: D, Đ, I, K, P, S, X (over 95%). 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 Vietnamese 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 without illumination) 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 cross line descriptors combined with invariant moments and they are used to train SVMs. The focus of this research is the resizing method and extracting features which bring high accuracy in classifying different gestures, 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 is presented in the Table 1 (with the same training and testing images).

Table I. Methods Comparison

Approach	***Our***	***[10]***	***[11]***	***[12]***	***[13]***	***[14]***
Recognition (%)	95	84	92.78	90.45	94.2	90.45

As further work, our method will be improved to recognize the gesture even when the hand inclination is large, or the hand is not directly opposite to the camera.

VI. 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).

VII. REFERENCES

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

[2] Soren Lenman, Lars Bretzner, Bjorn Thuresson, “Computer Vision Based Hand Gesture Interfaces for Human – Computer Interaction”, Department of Numerical Analysis and Computer Science, June 2002.

[3] M. Marshall, “Virtual Sculpture - Gesture Controlled System for Artistic Expression”, Proceedings of the AISB 2004 COST287 - ConGAS Symposium on Gesture, Interfaces for Multimedia Systems, Leeds, UK, 2004, pp. 58-63.

[4] 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, 2006, pp. 1-4.

[5] The Duy Bui and Long Thang Nguyen, “Recognition of Vietnamese sign language using MEMS accelerometers”, 1st International Conference on Sensing Technology, Palmerston North, New Zealand, November 2005.

[6] Utsumi A. and Ohya J., “Multiple Hand Gesture Tracking using Multiple Cameras”, Proc. Int. Conf. on Computer Vision and Pattern Recognition, 1999, pp.473–478.

[7] Ueda E., "A Hand Pose Estimation for Vision-Based Human Interfaces", IEEE Transactions on Industrial Electronics, Vol. 50, No. 4, 2003, pp. 676–684.

[8] Bettio, F. et al, "A Practical Vision-Based Approach to Unencumbered Direct Spatial Manipulation in Virtual Worlds", Eurographics Italian Chapter Conf., 2007.

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

[10] Mokhtar M. Hasan, Pramoud K. Mirsa, "Brightness Factor Matching For Gesture Recognition System Using Scaled Normalization", International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3(2), 2011.

[11] V. S. Kulkarni, S. D. Lokhande, "Appearance Based Recognition of American Sign Language Using Gesture Segmentation", International Journal on Computer Science and Engineering (IJCSE), Vol. 2(3), 2010, pp. 560-565.

[12] Shuying Zhao, Wenjun Tan, Shiguang Wen, and Yuanyuan 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, 2008, pp. 786–794.

[13] Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yang, Toshiaki Ejima, "Hand Gesture Recognition Using Hidden Markov Models", IEEE International Conference on computational cybernetics and simulation, Vol.5, 1997.

[14] E. Stergiopoulou, N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," Elsevier Engineering Applications of Artificial Intelligence, Vol. 22(8), 2009, pp. 1141 – 1158.

[15] Sebastien Marcell - Hand Posture and Gesture Datasets, www.idiap.ch/resource/gestures

[16] Thomas Moeslund's Gesture Recognition home page, www-prima.inrialpes.fr/FGnet/data/12-MoeslundGesture

[17] Fleck M., Forsyth D. and Bregler C., "Finding Naked People", European Conference on Computer Vision, 1996.

[18] Jonathan Milgram, Mohamed Cheriet and Robert Sabourin, "One Against One or One Against All: Which One is Better for Handwriting Recognition with SVMs?", 10th International Workshop on Frontiers in Handwriting Recognition, 2006.

[19] C. Bond., "An Efficient and Versatile Flood Fill Algorithm for Raster Scan Displays", 2011.

[20] Bharath K. Sriperumbudur, Kenji Fukumizu, Gert R. G. Lanckriet, "Learning in Hilbert vs. Banach Spaces: A Measure Embedding Viewpoint", NIPS, 2011, pp.1773-1781.

[21] T. Ahonen, A. Hadid, M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28 (12), Dec. 2006, pp. 2037-2041.

[22] Arpit Mittal, Andrew Zisserman and Philip Torr, "Hand detection using multiple proposals", Proceedings of the British Machine Vision Conference, September 2011, pp. 75.1-75.11.

[23] Mark Nixon and Alberto Aguado, *Feature Extraction & Image Processing 2nd*, Academic Press, UK, 2008.