International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:2, 2009

Real-Time Hand Tracking and Gesture Recognition System Using Neural Networks

Tin Hninn Hninn Maung
Department of Engineering Physics, Mandalay Technological University, Mandalay, Myanmar
tinhninhninmg@gmail.com

***Abstract*—**This paper introduces a hand gesture recognition system to recognize real time gesture in unstrained environments. Efforts should be made to adapt computers to our natural means of communication: Speech and body language. A simple and fast algorithm using orientation histograms will be developed. It will recognize a subset of MAL static hand gestures. A pattern recognition system will be using a transforrn that converts an image into a feature vector, which will be compared with the feature vectors of a training set of gestures. The final system will be Perceptron implementation in MATLAB. This paper includes experiments of 33 hand postures and discusses the results. Experiments shows that the system can achieve a 90% recognition average rate and is suitable for real time applications.

***Keywords*—**Hand gesture recognition, Orientation Histogram, Myanmar Alphabet Language, Perceptron network, MATLAB.

I. Introduction

GESTURE recognition is an area of active current research in computer vision. Body language is an important way of communication among humans, adding emphasis to voice messages or even being a complete message by itself. Thus, automatics posture recognition systems could be used for improving human-machine interaction. This kind of human-machine interfaces would allow a human user to control remotely through hand postures a wide variety of devices.[1]. Many approaches to gesture recognition have been developed. Pavlovic[2] present an extensive review of existing technique for interpretation of hand gestures. A large variety of techniques have been used for modeling the hand. An approache based on the 2-D locations of fingertips and palms was used by Davis and Shah [3]. Bobick and Wilson [4] have developed dynamic gestures have been handled using framework. A state-based technique for recognition of gestures in which the define the feature as a sequence of states in a measurement or configuration space. Human- computer interaction using hand gestures has been studied by a number of researchers like Starnes and Pentland [5], and Kjeldsen and Kender [6]. Use of learning for hand gesture recognition has been explored in [7]. Yoon et al.[9] have proposed a recognition scheme using combined features of location, angle and velocity. Lockton et al.[10] proposed a real time gesture recognition system, which can recognize 46 MAL, letter spelling alphabet and digits. The gestures that are recognized [10] are 'static gesture' of which the hand gestures do not move. Several system use Hidden Markov models for gesture recognition [8]. This research is focused on the application of the Neural Networks methods for hand gesture recognition. This paper is organized as follows: Section 2 introduces concept of neural networks and image data base approach is described in Section 3. Object recognition with large object tracking and shape recognition are represented in Section 4. The goal of the proposed system is in Section 5. Conclusions are finally described in Section 6.

II. Neural Networks

Neural networks have already proven efficient for classification, have been used and lead to very satisfying results: the rate of successful recognition can reach 95%. The difficulty is to perform the training and all the preprocessing it requires. Neural networks can be solved some of the problems in vision for which procedural algorithms are difficult to generate and we consider optical computing as a means to providing the greater computing power required for real time and more general- purpose vision systems.

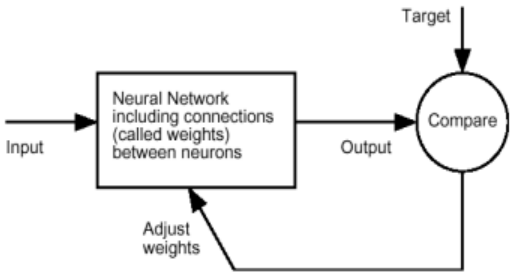

Fig. 1. Neural Net block diagram

There are a variety of benefits that an analyst realizes from using neural networks in their work.

a) Pattern recognition is a powerful technique for harnessing the information in the data and generalizing about it. Neural nets learn to recognize the patterns which exist in the data set.
b) The system is developed through learning rather than programming. Programming is much more time consuming for the analyst and requires the analyst to specify the exact behavior of the model. Neural nets teach themselves the patterns in the data freeing the analyst for more interesting work.
c) Neural networks are flexible in a changing environment. Rule based systems or programmed systems are limited to the situation for which they were designed - -when conditions change, they are no longer valid. Although neural networks may take some time to learn a sudden drastic change, they are excellent at adapting to constantly changing information.
d) Neural networks can build informative models where more conventional approaches fail. Because neural networks can handle very complex interactions they can easily model data which is too difficult to model with traditional approaches such as inferential statistics or programming logic.
e) Performance of neural networks is at least as good as classical statistical modeling, and better on most problems. The neural networks build models that are more reflective of the structure of the data in significantly less time.

III. Image Database

The starting point of the project was the creation of a data base with all the images that would be used for training and testing. The image data can have different formats. Images can be either hand drawn, digitized photographs or a 3D dimensional hand. Photographs were used, as they are the most realistic approach. Images came from two main sources. Various Myanmar Alphabet Language databases on the Internet and photographs that are taken with a digital camera. This meant that they have different sizes, different resolutions and some times almost completely different angles of shooting.

Two operations were carried out in all of the images. They were converted to grayscale and the background was made uniform. The internet databases already had uniform backgrounds but the ones is taken with the digital camera had to be processed in Adobe Photoshop. Drawn images can still simulate translational variances with the help of an editing program (e.g. Adobe Photoshop). The database itself was constantly changing throughout the completion of the project as it would decide the robustness of the algorithm. Therefore, it had to be done in such way that different situations could be tested and thresholds above which the algorithm didn't classify correct would be decided. The construction of such a database is clearly dependent on the application. If the application is a crane controller for example operated by the same person for long periods the algorithm doesn't have to be robust on different person's images. In this case noise and motion blur should be tolerable. For most of the gestures the training set originates from a single gesture. Those enhanced in Adobe Photoshop using various filters. The reason for this is to get the algorithm to be very robust for images of the same database.

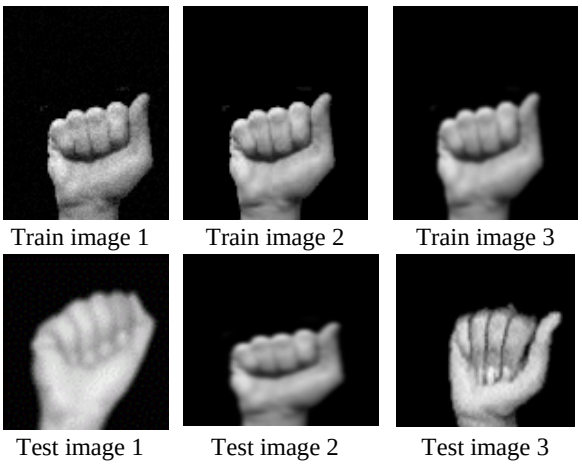

Fig. 2. A set of Train-Test images in different gesture stored in the image database.

IV. Object Recognition

a) Large Object Tracking

In some interactive applications, the computer needs to track the position or orientation of a hand that is prominent in the image. Relevant applications might be computer games, or interactive machine control. In such cases, a description of the overall properties of the image may be adequate. Image moments, which are fast to compute, provide a very coarse summary of global averages of orientation and position. If the hand is on a uniform background, Large Object Tracking method can distinguish hand positions and simple pointing gestures. The large-object -tracking method makes use of a low-cost detector/processor to quickly calculate moments. This is called the artificial retina chip. This chip combines image detection with some low-level image processing (named artificial retina by analogy with those combined abilities of the human retina). The chip can compute various functions useful in the fast algorithms for interactive graphics applications.

b) Shape recognition

Most applications, such as recognizing particular static hand signal, require the shape of the input object than image moments provide. If the hand signals fell in a predetermined set, and the camera views a close-up of the hand, combined with a simple method top analyze hand signals called

orientation histograms. These applications involve two phases; training and running. In the training phase, the user shows the system one or more examples of a specific hand shape. The computer forms and stores the corresponding orientation histograms. In the run phase, the computer compares the orientation histogram of the current image with each of the stored templates and selects the category of the closest match, or interpolates between templates, as appropriate. This method should be robust against small differences in the size of the hand but probably would be sensitive to changes in hand orientation.

V. Goals

The goals of this paper are to create a method to recognize hand gestures, based on a pattern recognition technique employing histograms of local orientation. The orientation histogram will be used as a feature vector for gesture classification and interpolation.

a) Orientation Histograms for Hand Gesture Recognition

We use the orientation histogram as a feature classification and interpolation. It has the advantage of being in lighting change condition. This method is simple and fast to compute and offers some robustness to scene illumination changes. It may provide a more natural human-computer interface, allowing people to point, or rotate a CAD model by rotating their hands. If the computer could understand player's hand gestures, interactive computer games would be enhanced and even useful to control household appliances. For the broadest possible application, a gesture recognition algorithm should be fast to compute. Applying a simple pattern recognition method to hand gesture recognition, resulting in a fast, usable hand gesture recognition system. We apply some transformation T to the image data to form a feature which represent that particular gesture. To classify the gesture, we compare the feature vectors from a previously generated training set. For static hand gestures, we use the histogram of local orientation as a feature vector for recognition. We have applied a simple pattern recognition technique to the problem of hand gesture recognition. This method has a training phase and a run phase. In the training phase, the user shows example hand gesture commands. The computer stores one or more feature vectors, blurred orientation histograms for each command. In the run phase, the computer compares the feature vector for the present image with those in the training set and picks the category of the nearest vector, or interpolates between vectors. The methods are image-based, simple and fast. We have implemented a real-time version, using an ordinary workstation with no special hardware beyond a digital camera.

b) Operation

The program can be 'divided' in 6 steps. Let's examine them one by one.

i) Step1

The first thing for the program to do is to read the image database. Single for loop is used to read an entire folder of images and store them in MATLAB's memory. The folder is selected by the user from menus. A menu will firstly pop-up asking you whether you want to run the algorithm on test or train sets. Then a second menu will pop-up for the user to choose which MAL sign he wants to use.

ii) Step2

Resize all the images that were read in Step1 to 150x140 pixels. This size seems the optimal for offering enough detail while keeping the processing time low.

iii) Step3.

Next thing to do is to find the edges. As mentioned before 2 filters were used.

For the x direction x = [0 -1 1]

For the y direction $y = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$ which is the same as x but transposed and multiplied by -1.

As this would meant the only feature extracted from the images it had to offer enough discrimination among them. From the images above it doesn't seem like a good edge detector. In fact it doesn't look like an edge detector that much. During the beginning of the algorithm's writing I have applied with all the known edge detectors (Sobel, Prewitt, Roberts and Canny). I have also tried various line detectors and combination of those two. An interesting way of testing various edge detectors and changing their values is an Adobe Photoshop filter which lets you input the values of the mask in a matrix and see the result on the image in real time. Even though some combinations of Canny (or Sobel) with line detection algorithms produced a very good result – that is visual- it wouldn't offer to much differentiation between the images. Of course for all those operators threshold values and some other parameters can be set as well. Therefore there was a lot of combinations to be tried but as mentioned before either the differentiation wasn't enough or in extreme conditions it was too much. In this case every image would classify in a class of each own or at all.

iv) Step 4

Dividing the two resulting matrices (images) dx and dy element by element and then taking the arctan ($\tan^{-1}$). This will give the gradient orientation.

v) Step 5

Then the MATLAB function im2col is called to rearrange the image blocks into columns. This is not a necessary step but it has to be done if we want to display the orientation histogram. An angle histogram, which is a polar plot showing the distribution of values grouped according to their numeric range. While developing the algorithm those histograms are the fastest way of getting a good idea how good the detection is done.

vi) Step 6

Converting the column matrix with the radian values to degrees. This way we can scan the vector for values ranging from 0^o to 90^o .This is because for real elements of X,atan(X)

is in the range .This can also see from the orientation histograms where values come up only on the first and last quarter. Determining the number of the histogram bins was another issue that was solved by experimenting with various values. At the same time the neural network is thought itself as this vector would be the input to the network. The smaller the vector, the faster the processing. The algorithms development was organized having in mind MATLAB weaknesses. The major one is speed. MATLAB is perfect for speeding up the development process but it can be very slow on execution when bad programming practices have been employed. Nested loops slow down the program considerably. It is probably because MATLAB is built on loops. Therefore, unnecessary back-tracking was avoided and even some routines were written in full instead of using for loops. The code is not much in any case.

VI. Perceptron implementation in MATLAB

We will apply a learning rule as a procedure for modifying the weights and biases of a network. (This procedure may also be referred to as a training algorithm.) The learning rule is applied to train the network to perform some particular task. Learning rules in the MATLAB toolbox fall into two broad categories: supervised learning and unsupervised learning. The algorithm has been developed using supervised learning. In supervised learning, the learning rule is provided with a set of examples (the training set) of proper network behavior: where is an input to the network, and is the corresponding correct (target) output. As the inputs are applied to the network, the network outputs are compared to the targets. The learning rule is then used to adjust the weights and biases of the network in order to move the network outputs closer to the targets. The perceptron learning rule falls in this supervised learning category.

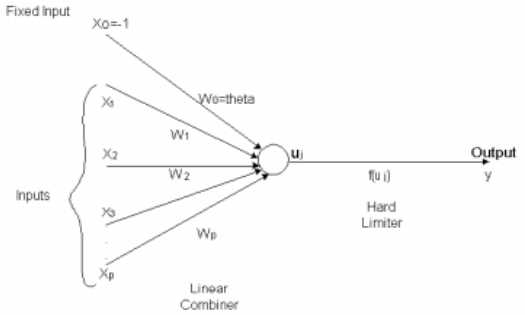

Fig. 3. Perceptron single flow graph

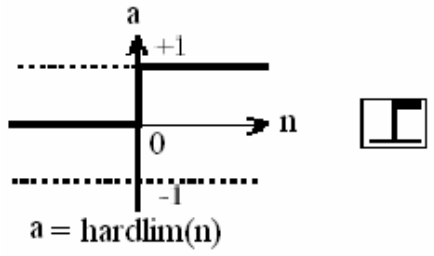

Fig. 4. Hard-limit transfer function

This is the hard limit function used for the Perceptron algorithm written in MATLAB to create neurons that make a classification decision.

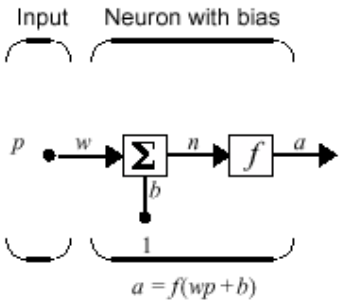

Fig. 5. Neuron with singular scalar input

VII. Conclusions

We have developed a gesture recognition system that is proved to be robust for Myanmar Alphabet Languages gestures. The system is fully automatic and it works in real-time. It is fairly robust to background cluster. The advantage of the system lies in the ease of its use. The users do not need to wear a glove, neither is there need for a uniform background. Experiments on a single hand database have been carried out and recognition accuracy of up to 98% has been achieved. We plan to extend our system into 3D tracking. Currently, our tracking method is limited to 2D. We will therefore investigate a practice 3D hand tracking technique using multiple cameras. Focus will also be given to further improve our system and the use of a lager hand database to test system and recognition. The idea of the project got started from a McConnel's idea of orientation histograms and trying to use this technique in conjunction with Neural Networks. Pattern recognition of orientation histograms have been used different ways of comparing and classifying were employed. It is efficient as long as the data sets are small and not further improvement is expected. Another advantage of using neural networks is that you can draw conclusions from the network output. If a vector is not classified we can check its output and work out a solution. My major goal was speed and the avoidance of special hardware. This was achieved although it would be faster if written C/C++ but the complexity of the design and implementation would have been much higher. MATLAB is slower but allows its users to work faster and concentrate on the results and not on the design. 12]

References

[1] Journal of WSCG, Vol.12, NO 1-3, ISSN 1213-6972
[2] V.I. Pavlovic, R. Sharma, T.S. Huang. Visual interpretation of hand gestures for human-computer interaction, A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7): 677-695, 1997.
[3] J.Davis, M.Shah. Recognizing hand gestures. In Proceedings of European Conference on Computer Vision, ECCV: 331-340, 1994.
[4] D.J.Turman, D. Zeltzer. Survey of glove-based input. IEEE Computer Graphics and Application 14:30-39, 1994

[5] Starner, T. and Pentland. Real-Time American Sign Language Recognition from Video Using Hidden Markov Models, TR-375, MIT Media Lab, 1995.
[6] R.Kjeldsen, J.Kender. Visual hand gesture recognition for window system control, in IWAFGR: 184-188, 1995.
[7] M.Zhao, F.K.H. Quek, Xindong Wu. Recursive induction learning in hand gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11): 1174.
[8] Hyeon-Kyu Lee, Jin H. Kim. HMM_based threshold model approach for gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell.201(10):961-973,1999.
[9] Ho-Sub Yoon,Jung Soh,Younglae J. Bae, Hyun Seng Yang. Hand Gesture recognition using combined features of location, angle, velocity, Pattern Recognition 34:1491-1501,2001.
[10] R.Locken, A.W. Fitzgibbon. Real gesture recognition using deterministic boosting , Proceeding of British Machine Vision Conference, 2002.
[11] Hand Gesture Recognition Using Neural Networks by Klimis Symeonidis.
[12] Duane Hanselman, Bruse Littlefield, Mastering MatLab, A comprehensive tutorial and reference, Prentice Hall.