Instrumentation and Measurement
Technology Conference – IMTC 2007
Warsaw, Poland, May 1-3, 2007

Real-time Vision-based Hand Gesture Recognition Using Haar-like Features

Qing Chen, Nicolas D. Georganas, Emil M. Petriu

DISCOVER Lab
School of Information Technology and Engineering
University of Ottawa
800 King Edward Ave. Ottawa, Ontario, Canada K1N 6N5
Phone: +1 613 5625800X2148, Fax: +1 613 5625664
Email: {qchen, georganas, petriu}@discover.uottawa.ca

Abstract *– This paper proposes a two level approach to solve the problem of real-time vision-based hand gesture classification. The lower level of the approach implements the posture recognition with Haar-like features and the AdaBoost learning algorithm. With this algorithm, real-time performance and high recognition accuracy can be obtained. The higher level implements the linguistic hand gesture recognition using a context-free grammar-based syntactic analysis. Given an input gesture, based on the extracted postures, the composite gestures can be parsed and recognized with a set of primitives and production rules.*

Keywords *– posture, gesture, Haar-like features, AdaBoosting, grammar.*

I. INTRODUCTION

Human-computer interfaces (HCI) have evolved from text-based interfaces through 2D graphical-based interfaces, multimedia-supported interfaces, to full fledged multi-participant Virtual Environment (VE) systems. While providing a new sophisticated paradigm for human communication, interaction, learning and training, VE systems also provide new challenges since they include many new types of representation and interaction. The traditional two-dimensional HCI devices such as keyboards and mice are not enough for the latest VE applications. Instead, VE applications require utilizing several different modalities and technologies and integrating them into a more immersive user experience [1]. Devices that sense body position and hand gestures, speech and sound, facial expression, haptic response and other aspects of human behavior or state can be used so that the communication between the human and the VE can be more natural and powerful.

To achieve natural human-computer interaction for VE applications, the human hand could be considered as an input device. Hand gestures are a powerful human to human communication modality. However, the expressiveness of hand gestures has not been fully explored for HCI applications. Compared with traditional HCI devices, hand gestures are less intrusive and more convenient to explore the three-dimensional (3D) virtual worlds [2].

The human hand is a complex articulated object consisting of many connected parts and joints. Considering the global hand pose and each finger joint, human hand motion has roughly 27 degree of freedom (DOF) [2]. To use human hands as a natural HCI, glove-based devices such as the CyberGlove have been used to capture human hand motions. However, the gloves and their attached wires are still quite cumbersome and awkward for users to wear, and moreover, the cost of the glove is often too expensive for regular users. With the latest advances in the fields of computer vision, image processing and pattern recognition, real-time vision-based hand gesture classification is becoming more and more feasible for human-computer interaction in VEs. Early research on vision-based hand tracking usually needs the help of markers or colored gloves to make the image processing easier. In current state-of-the-art vision-based hand tracking and gesture classification, the research is more focused on tracking the bare hand and recognizing hand gestures without help of any markers and gloves. Meanwhile, the vision-based hand gesture recognition system also needs to meet the requirements including real-time performance, accuracy, and robustness.

Vision-based hand gesture recognition techniques can be divided into two categories: appearance-based approaches and 3D hand model-based approaches [3]. Appearance-based approaches use image features to model the visual appearance of the hand and compare these parameters with the extracted image features from the input video. 3D hand model-based approaches rely on a 3D kinematic hand model with considerable degrees of freedom and try to estimate the hand parameters by comparison between the input images and possible 2D appearance projected by the 3D hand model. Generally speaking, appearance-based approaches have the advantage of real-time performance due to the easier 2D image features employed. 3D hand model-based approaches offer a rich description that potentially allows a wide class of hand gestures. However, as the 3D hand models are articulated deformable objects with many degrees of freedom, a very large image database is required to cover all the characteristic shapes under different views. Matching the query image frames from video input with all images in the database is time-consuming and computationally expensive. Another problem is lack of the capability to deal with singularities that arise from ambiguous views [4].

II. TWO LEVEL APPROACH

In the literature of gesture recognition, there are two important definitions need to be cleared: hand posture and hand gestures. A hand posture is defined solely by the static hand configuration and hand location without any movements involved. A hand gesture refers to a sequence of hand postures connected by continuous motions (global hand motion and local finger motion) over a short time span. A hand gesture is a composite action constructed by a series of hand postures that act as transition states. With this composite property of hand gestures, it is natural to decouple the problem of gesture recognition into two levels – low level posture recognition and high level gesture recognition. For hand postures, the repeatability is usually poor due to the high degree of freedom of the hand as well as the difficulty of duplicating the same working environment such as the background and the lighting condition. To solve the problem, we use a statistical approach based on a set of Haar-like features that focus more on the information within a certain area of the image rather than each single pixel. To improve the classification accuracy and achieve the real-time performance, we employ the AdaBoost (Adaptive Boost) learning algorithm that can adaptively select the best features in each step and combine them into a strong classifier. The training algorithm based on AdaBoost learning algorithm takes a set of "positive" samples, which contain the object of interest (in our case: hand postures), and a set of "negative" samples, *ie*. images do not contain objects of interest. During the training process, distinctive Haar-like features are selected to classify the images containing the object of interest at each stage [5].

The statistical approach can describe the posture quantitatively using numeric parameters. However, the quantitative description is not adequate to represent a hand gesture's hierarchical structure. Under this situation, a syntactic object description is more appropriate to represent the structure of hand gestures [6]. With a grammar-based approach to convey the hierarchical nature of hand gestures, we can construct a concrete representation for the hand gestures, and thus enables the system to recognize the gestures based on a set of primitives and production rules.

III. POSTURE RECOGNITION WITH HAAR-LIKE FEATURES

Originally for the task of face tracking and detection, Viola and Jones employed a statistical approach to handle the large variety of human faces [7]. In their algorithm, the concept of "integral image" is used to compute a rich set of Haar-like features. Compared with other approaches, which must operate on multiple image scales, the integral image can achieve true scale invariance by eliminating the need to compute a multi-scale image pyramid, and significantly reduces the image processing time. Another technique used by this approach is the feature selection algorithm based on the AdaBoost learning algorithm. The Viola and Jones algorithm is approximately 15 times faster than any previous approaches while achieving equivalent accuracy as the best published results [7].

The simple Haar-like features (so called because they are computed similarly to the coefficients in the Haar wavelet transform) are used in the Viola and Jones algorithm. There are two motivations for the employment of the Haar-like features rather than raw pixel values. The first reason is that the Haar-like features can encode ad-hoc domain knowledge, which is difficult to describe using a finite quantity of training data. Compared with raw pixels, the Haar-like features can efficiently reduce/increase the in-class/out-of-class variability and thus making classification easier [8]. The Haar-like features describe the ratio between the dark and bright areas within a kernel. One typical example is that the eye region in the human face is darker than the cheek region, and one Haar-like feature can efficiently catch that character. The second motivation is that a Haar-like feature-based system can operate much faster than a pixel-based system. Besides the above advantages, the Haar-like features are also relatively robust to noise and lighting changes because they compute the gray level difference between the white and black rectangles. The noise and lighting variations affect the pixel values on the whole feature area, and this influence can be counteracted.

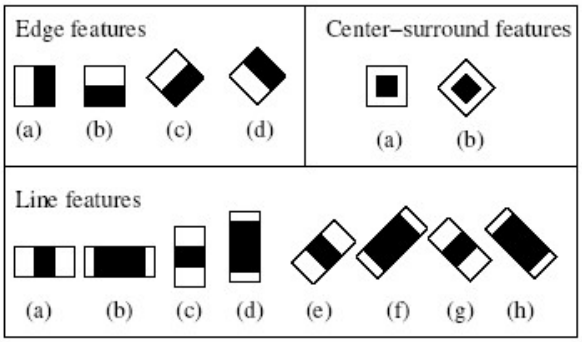

Fig.1. A set of Haar-like features

Each Haar-like feature consists of two or three connected "black" and "white" rectangles. Fig. 1 shows the extended Haar-like features set proposed by Lienhart and Maydt [8]. The value of a Haar-like feature is the difference between the sums of the pixel values within the black and white rectangles.

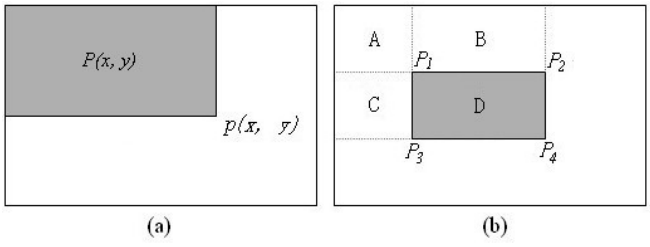

Fig.2. The concept of "integral image"

The "integral image" at location of *pixel(x, y)* contains the sum of the pixel values above and left of this pixel inclusive (see Fig. 2 (a)):

$$P(x, y) = \sum_{x' \le x, y' \le y} p(x' y')$$

According to the definition of "integral image", the sum of the pixel value within the area *D* in Fig. 2 (b) can be computed by:

$$P_1 + P_4 - P_2 - P_3$$

where P_1 = A, P_2 = A+B, P_3 = A+C, and P_4 = A+B+C+D as per Fig. 2(b).

Fig.3. Detecting a human face with a Haar-like feature

To detect an object of interest, the image is scanned by a sub-window containing a specific Haar-like feature (see the face detection example in Fig. 3). Based on each Haar-like feature f_j, a correspondent classifier $h_j(x)$ is defined by:

$$h_j(x) = \begin{cases} 1, \ if \ \ p_j f_j(x) < p_j \theta_j \\ 0, \quad otherwise \end{cases}$$

where x is a sub-window, and θ is a threshold. p_j indicates the direction of the inequality sign.

In practice no single Haar-like feature can detect the object with a very high accuracy. However, it is not difficult to achieve a series of weak classifiers with the accuracy slightly better than 50% using Haar-like features. The AdaBoost learning algorithm is a method to improve the accuracy based on a series of weak classifiers stage by stage [9]. The AdaBoost learning algorithm initially maintains a uniform distribution of weights over each training samples. In the first iteration, the algorithm trains a weak classifier using one Haar-like feature that achieves the best recognition performance for the training samples. In the second iteration, the training samples that were misclassified by the first weak classifier receive higher weights so that the newly selected Haar-like feature must focus more computation power towards these misclassified samples. The iteration goes on and the final result is a cascade of linear combinations of the selected weak classifiers, *i.e.* a strong classifier, which achieves the required accuracy (see Fig. 4).

In practical implementation, the attentional cascade is employed to speed up the performance of the learning algorithm. In the first stage of the training process, the threshold of the weak classifier is adjusted low enough so that 100% of the target objects can be detected while keeping the false negative rate close to zero [10]. The trade-off of a low threshold is that a higher false positive detection rate will accompany. A positive result from the first classifier triggers the evaluation of a second classifier, which has also been adjusted to achieve very high detection rates. A positive result from the second classifier triggers a third classifier, and so on.

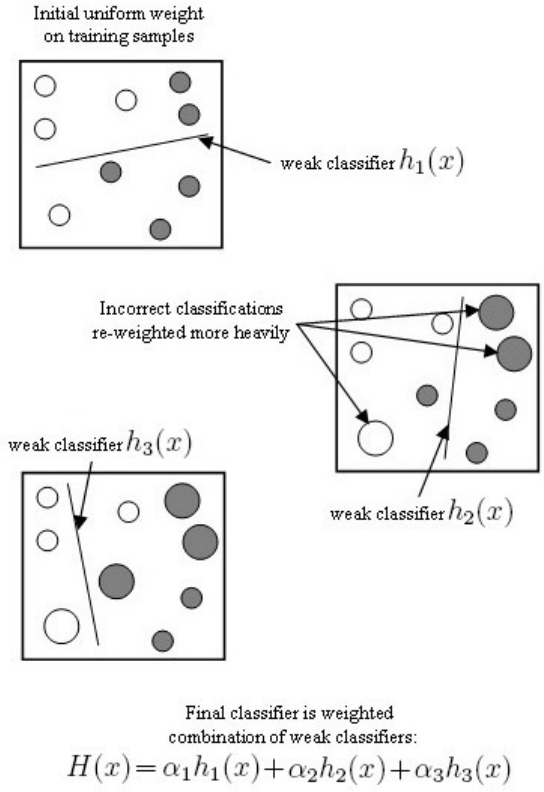

Fig.4. The AdaBoost learning algorithm

To be detected by the trained cascade, the positive sub-windows must pass each stage of the cascade. A negative outcome at any point leads to the immediate rejection of the sub-window (see Fig. 5).

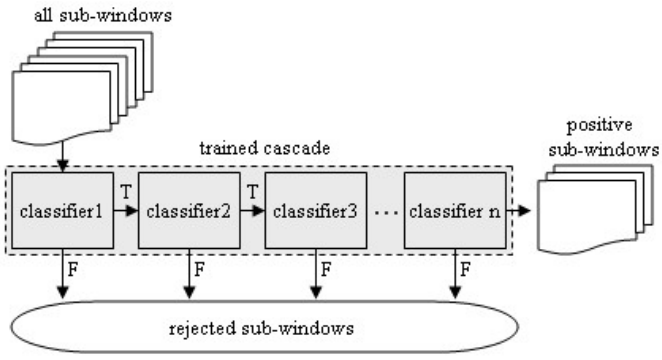

Fig.5. Detection of positive sub-windows using the cascade

The reason for this strategy is based on the fact that the majority of the sub-windows are negative within a single image frame, and it is a rare event for a positive sub-window to go through all of the stages. With this strategy, the cascade can significantly speed up the processing time as the initial weak classifiers try to reject as many negative sub-windows as possible and more computation power will be focused on the more difficult sub-windows that passed the scrutiny of the initial stages of the cascade.

In our implementation, four hand postures are tested: the two-finger posture, the palm posture, the fists posture and the little finger posture (see Fig. 6). The camera used for the video input in our experiment is a low-cost Logitech QuickCam web-camera that provides video capture with the resolution of 320×240, 15 frames-per-second.

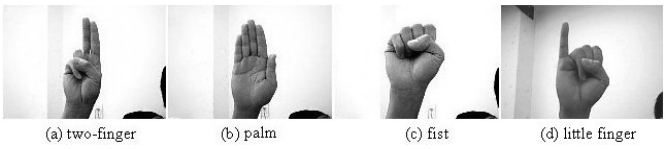

Fig.6. Four tested postures

Fig.7. Part of the positive samples of the “two-finger” posture

We collected the positive samples from a user’s hand for the preliminary experiment. The experiment and testing were implemented in the laboratory with natural fluorescent lighting condition. To change the illumination condition, we installed an extra incandescent light bulb to create a tungsten lighting condition. We collected around 450 samples with different scales for each posture. To increase the robustness of the classifier, we intentionally included a number of positive samples with certain in-plane rotations and out-of-plane rotations. Fig. 7 shows some positive samples of the ``two-finger" posture. To simplify the task at the initial stage of the experiment, we keep the white wall as the background.

500 random images that do not have the hand postures are collected as the negative samples for the training process. Fig. 8 shows some negative samples. All negative samples are passed through a background description file which is a text file containing the filenames (relative to the directory of the description file) of all negative sample images.

With the required false alarm rate set at 1×10^{-6}, a 15-stage cascade classifier is obtained for the “two-finger” posture, and the true positive detection rate is 97.5%. For the “palm” posture, a 10-stage cascade classifier is obtained with a true positive detection rate at 98%. For the “fist” posture, a 15-stage cascade classifier is obtained with a true positive detection rate at 98%. For the “little finger” posture, a 14-stage cascade classifier is obtained with a true positive detection rate at 97.1%. In order to evaluate the performance of the obtained classifiers, 100 testing images for each posture are collected separately with similar backgrounds but different illumination conditions. Table 1 shows the performance of these cascade classifiers.

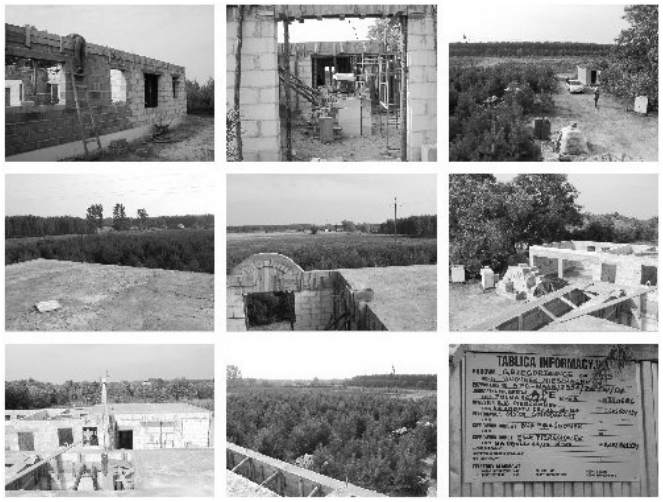

Fig.8. Part of the negative samples used in the training process

Table 1. The performance of the trained classifiers

Posture Name	Hits	Missed	False	Detection time (second)
Two-finger	100	0	29	3.049000
Palm	90	10	0	1.869000
Fist	100	0	1	2.829000
Little	93	7	2	2.452000

By analyzing the detection results (see Fig. 9), we found that some of the missed positive pictures are caused by the excessive in-plane or out-of-plane rotations. For the false detections, the majority of them only happened in very small image areas, which have a higher probability of containing similar bright/dark patterns. These small false detection boxes can be easily eliminated by defining a threshold for the rectangular size. The maximum time required to detect all 100 testing images is 3.049 seconds for the “`two-finger” gesture classifier. The time required for the rest classifiers are

all within 3 seconds. We tested the real-time performance with live input from the web-camera, and there is no detectable pause and latency to track and detect the hand postures with all our trained classifiers. The trained classifiers showed a certain degree of robustness against the in-plane rotation as long as the rotation range is within the scope of ±15º. The classifiers also showed a certain degree of tolerance for out-of-plane rotation and pretty good robustness against lighting variance.

Fig.9. Some detection results for the "two-finger" posture with the trained cascade classifier.

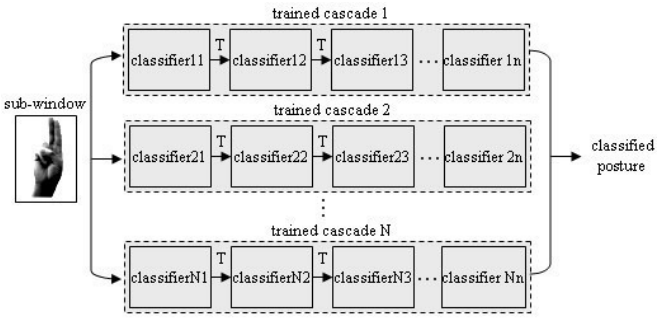

Fig.10. The parallel cascades structure for hand posture classification

With high detection speed, we implemented a parallel cascades structure to classify different gestures (see Fig. 10). In this structure, multiple cascades are loaded into the system simultaneously, and each cascade is responsible to detect a single hand posture. Rectangles of different gray levels are used to tell which posture is detected. Based on the experimental results, we found the real-time performance is not impaired when we load all four trained cascade classifiers at the same time. Confusions do occur between the "fist posture" and the "little finger" posture as well as the "two-finger" posture (see Fig. 11). However, these confusions can be resolved by the grammar-based syntactic analysis with a set of appropriate primitives.

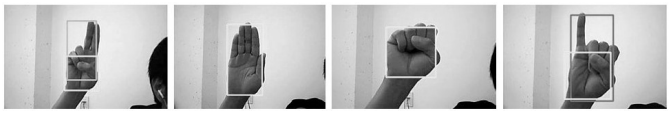

Fig.11. Examples of recognized postures

IV. A CONTEXT-FREE GRAMMAR FOR GESTURE RECOGNITION

The high level recognition of gestures requires an appropriate grammar that can describe the complex gestures with simple primitives and rules. In pattern recognition, a formal grammar is defined by:

$$G = (N, T, P, S)$$

where N is a finite set of *non-terminal symbols*, T is a finite set of *terminal symbols* that is disjoint from N, P is a finite set of *production rules*, and S is a *start symbol* $\in N$. According to [6], the principles to identify the primitives include:

1. The number of primitive types should be small.
2. The primitives selected must be able to form an appropriate object representation.
3. Primitives should be easily segmentable from the image.
4. Primitives should be easily recognizable using some statistical pattern recognition method.
5. Primitives should correspond with significant natural elements of the object structure being described.

For the description of hand gestures, the parallel cascade structure used by us effectively meet the requirements of item 3 and item 4 if we define the different gray level rectangles as terminal symbols and the postures as non-terminal symbols for gestures. After the primitives have been successfully extracted, an appropriate grammar representing a set of production rules that must be given to construct different hand gestures.

A context-free grammar is a grammar in which the left-hand side of each production rule consists of only a single non-terminal symbol, and the right-hand side is a string consisting of terminals and/or non-terminals. The term "context-free" expresses the fact that the non-terminal symbol can always be replaced by the right-hand string, regardless of the context in which it occurs. Context-free grammars are powerful enough to describe the syntax of the hand gestures; On the other hand, context-free grammars are simple enough to allow the construction of efficient parsing algorithms which, for a given hand gesture, determine whether and how it can be generated from the grammar.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a two level approach to recognize hand gestures in real-time with a single web-camera as the input device. The low level of the approach is focused on the posture recognition with Haar-like features and the AdaBoost learning algorithm. The Haar-like features can effectively describe the hand posture pattern with the computation of "integral image". The AdaBoost learning algorithm can efficiently speed up the performance speed and construct a strong classifier by combining a sequence of weak classifiers. Based on the cascade classifiers, a parallel cascade structure is implemented to classify different hand postures. From the experiment results, we find this structure can achieve satisfactory real-time performance as well as very high classification accuracy above 90%. For the high level hand gestures recognition, we proposed the context-free grammar to analyze the syntactic structure based on the detected postures.

In future work, we will implement the parsing algorithm for the context-free grammar. With this parsing algorithm, given an input gesture, we can determine its grammatical structure with respect to the given grammar so that the input gesture can be recognized.

REFERENCES

[1] M. Turk, *Gesture Recognition in Handbook of Virtual Environment Technology*, Lawrence Erlbaum Associates, Inc., 2001.

[2] Y. Wu, T. S. Huang, "Hand modeling analysis and recognition for vision-based human computer interaction," IEEE Signal Processing Magazine, Special Issue on Immersive Interactive Technology, Vol. 18, No. 3, pp. 51-60, 2001.

[3] H. Zhou, T. S. Huang, "Tracking articulated hand motion with Eigen dynamics analysis," *Proc. of International Conference on Computer Vision*, Vol. 2, pp. 1102-1109, 2003.

[4] D. D. Morris and J. M. Rehg. "Singularity analysis for articulated object tracking," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 289-296, 1998.

[5] G. Bradski, A. Kaehler, V. Pisarersky, "Learning-based computer vision with Intels open source computer vision library," *Intel Technology Journal*, Vol. 9, No. 2, pp. 119-130, 2005

[6] M. Sonka, V. Hlavac, R. Boyle, *Image processing, analysis, and machine vision*, PWS Publishing, 1999.

[7] P. Viola, M. Jones, "Robust real-time object detection," *Cambridge Research Laboratory Technical Report Series CRL2001/01*, pp. 1-24, 2001.

[8] R. Lienhart, J. Maydt, "An extended set of Haar-like features for rapid object detection," *Proc. IEEE International Conference on Image Processing ICIP 2002*, Vol. 1, pp. 900-903, 2002.

[9] Y. Freund, R. E. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, Vol. 14, No. 5, pp. 771-780, 1999.

[10] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE CVPR01*, pp. 511-518, 2001.