

Hand Gesture Recognition Using Haar-Like Features and a Stochastic Context-Free Grammar

Qing Chen, Nicolas D. Georganas, *Fellow, IEEE*, and Emil M. Petriu, *Fellow, IEEE*

Abstract—This paper proposes a new approach to solve the problem of real-time vision-based hand gesture recognition with the combination of statistical and syntactic analyses. The fundamental idea is to divide the recognition problem into two levels according to the hierarchical property of hand gestures. The lower level of the approach implements the posture detection with a statistical method based on Haar-like features and the AdaBoost learning algorithm. With this method, a group of hand postures can be detected in real time with high recognition accuracy. The higher level of the approach implements the hand gesture recognition using the syntactic analysis based on a stochastic context-free grammar. The postures that are detected by the lower level are converted into a sequence of terminal strings according to the grammar. Based on the probability that is associated with each production rule, given an input string, the corresponding gesture can be identified by looking for the production rule that has the highest probability of generating the input string.

Index Terms—Boosting, Haar-like features, hand gesture, hand posture, stochastic context-free grammar (SCFG).

I. INTRODUCTION

HUMAN-COMPUTER interfaces (HCIs) have evolved from text-based interfaces through 2-D graphical-based interfaces, multimedia-supported interfaces, to full-fledged multiparticipant 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 2-D 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 in exploring the 3-D 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, the human hand motion has roughly 27 degrees of freedom (DOFs) [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 the 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 the 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 3-D 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 video input. Three-dimensional hand model-based approaches rely on a 3-D kinematic hand model with considerable DOFs and try to estimate the hand parameters by comparison between the input images and the possible 2-D appearance projected by the 3-D hand model. Generally speaking, appearance-based approaches have the advantage of real-time performance due to the easier 2-D image features that are employed. Three-dimensional hand model-based approaches offer a rich description that potentially allows a wide class of hand gestures. However, as the 3-D hand models are articulated deformable objects with many DOFs, a very large image database is required to cover all the characteristic shapes under different views. Matching the query image frames from the video input with all the images in the database is time consuming and computationally expensive. Another problem is the lack of capability to deal with singularities that arise from ambiguous views [4].

Most of the current recognition techniques consider hand gestures as basic elements and analyze them without further breaking them into lower composite elements that would be

Manuscript received July 4, 2007; revised November 13, 2007.

The authors are with the DiscoverLab, School of Information Technology and Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: qchen@discover.uottawa.ca; georganas@discover.uottawa.ca; petriu@discover.uottawa.ca).

Digital Object Identifier 10.1109/TIM.2008.922070

easier to process. This results in rather low-speed, inaccurate, and fragile performance that is unsuited for real-time applications. Attempts were made to overcome this drawback by using markers or colored gloves [5], [6]. However, these techniques are still too cumbersome for vision-based hand tracking and gesture recognition in a robust, accurate, and easily accessible manner.

II. TWO-LEVEL APPROACH

In the literature of hand gesture recognition, there are two important definitions that need to be cleared [7], [8].

- *Hand posture*. A hand posture is a static hand pose and hand location without any movements involved.
- *Hand gesture*. A hand gesture refers to a sequence of hand postures that are connected by continuous motions over a short time span with the intent to convey information or interact with computers.

The dynamic aspect of the hand gesture includes global hand motions and local finger motions [9]. A hand gesture is a composite action that is 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—the low-level hand posture detection and the high-level hand gesture analysis.

The skin color and the hand shape are image features that are frequently used for hand posture detection [10]–[12]. Nevertheless, color-based algorithms face the difficult task of distinguishing objects such as the human arm and the face, which have similar color with the hand. To solve this problem, users are required to wear long-sleeve shirts, and restrictions are imposed on the colors of other objects in the observed scene. Color-based algorithms are also very sensitive to lighting variations. When the lighting does not meet the special requirements, the color-based algorithms usually fail. For shape-based algorithms, global shape descriptors such as Zernike moments and Fourier descriptors are used to represent the hand shape [11]. Most shape descriptors are pixel based, and the computational cost is usually too high to implement real-time systems. Another disadvantage is the requirement of noise-free image segmentation, which is a difficult task for the usually cluttered background images.

Considering the problems faced by color/shape-based approaches, as well as the poor repeatability for hand postures due to high DOFs of the hand and the difficulty to duplicate the same working environment such as backgrounds and lighting conditions, we decided to employ a statistical approach based on a set of Haar-like features, which focus more on the information within a certain area of the image rather than each single pixel. To improve classification accuracy and achieve real-time performance, we use the Adaptive Boost (AdaBoost) 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, i.e., images that 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 [13].

The statistical approach can quantitatively describe the hand posture using numeric parameters. However, the quantitative description is not adequate to represent a hand gesture’s structural information. In this situation, a syntactic object description is more appropriate to represent the composite characters of hand gestures [14]. 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, enable the system to recognize the gestures based on a set of primitives and production rules.

III. POSTURE DETECTION USING HAAR-LIKE FEATURES

Originally for the task of face tracking and detection, Viola and Jones [15] proposed a statistical approach to handle the large variety of human faces. 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 multiscale image pyramid and significantly reduces the image processing time. Another technique that is 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 accuracy that is equivalent to the best published results [15].

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 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, thus making the classification easier [16]. The Haar-like features describe the ratio between the dark and bright areas within a kernel. One typical example is that the eye region on the human face is darker than the cheek region, and one Haar-like feature can efficiently catch that characteristic. 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.

Each Haar-like feature consists of two or three connected “black” and “white” rectangles. Fig. 1 shows the extended Haar-like features set that was proposed by Lienhart and Maydt [16]. The value of a Haar-like feature is the difference between the sums of the pixel values in the black and white rectangles, i.e.,

$$f(x) = \sum_{\text{black}} (\text{pixel value}) - \sum_{\text{white}} (\text{pixel value}).$$

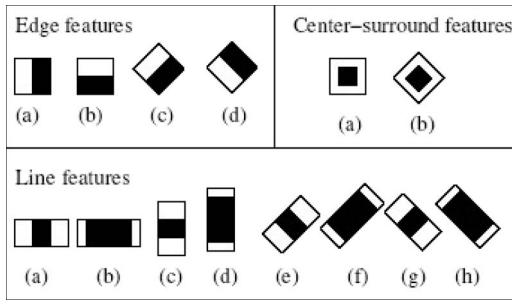


Fig. 1. Extended set of Haar-like features.

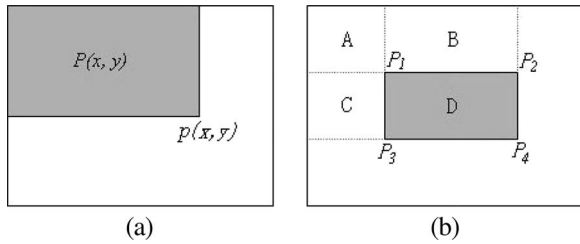


Fig. 2. Concept of the “integral image.”

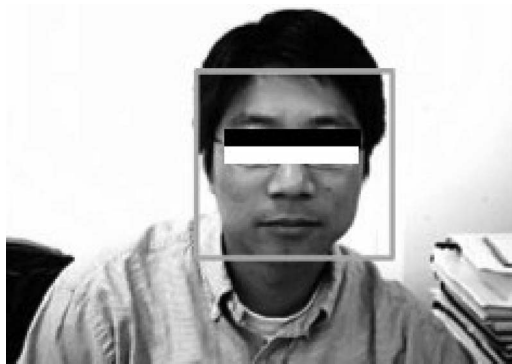


Fig. 3. Detecting a face with a subwindow containing a Haar-like feature.

The “integral image” at the location of pixel (x, y) contains the sum of the pixel values above and left of this pixel, which is inclusive [see Fig. 2(a)], i.e.,

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

According to the definition of the “integral image,” the sum of the pixel values 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 $P_4 = A + B + C + D$.

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

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

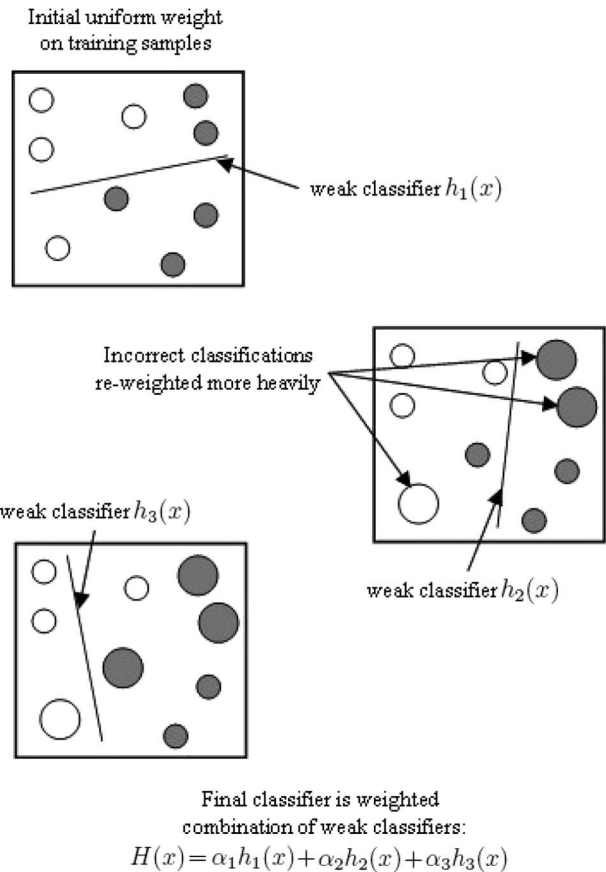


Fig. 4. Iteration of the AdaBoost learning algorithm.

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

In practice, no single Haar-like feature can identify the object with high accuracy. However, it is not difficult to find one Haar-like feature-based classifier that has better accuracy than random guessing. The AdaBoost learning algorithm can considerably improve the overall accuracy, stage by stage, by using a linear combination of these individually weak classifiers [17]. It should be noted that a Haar-like feature could be repeatedly used in the linear combination. The AdaBoost learning algorithm initially assigns an equal weight to each training sample (see Fig. 4). We start with the selection of a Haar-like feature-based classifier for the first stage, retaining the first one that yields better than 50% classification accuracy. This classifier is added to the linear combination with strength that is proportional to the resulting accuracy. For the next stage, the training samples are reweighted; training samples that are missed by the previous classifier are “boosted” in importance. The next classification stage must achieve better accuracy for these misclassified training samples so that the error can be reduced. We retain the classifier that further improves the overall classification accuracy. The iteration goes on by adding new classifiers to the linear combination until the overall accuracy meets the required level. The final result is a strong classifier composed of a cascade of the selected classifiers.

In practical implementation, an attentional cascade that was proposed by Viola and Jones [18] is used to speed up the process. At the first stage of the training process, the threshold

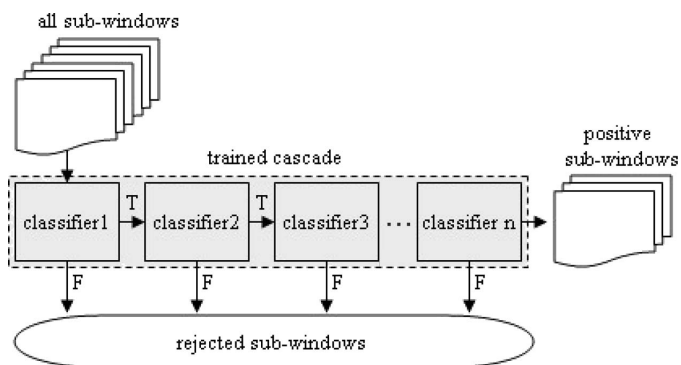


Fig. 5. Detection of positive subwindows using the trained cascade.

of the weak classifier is adjusted low enough so that 100% of the target objects can be detected. The tradeoff of a low threshold is that a higher false-positive detection rate will accompany the 100% true-positive detection rate. 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. To be detected by the trained cascade, the positive subwindows must pass each stage of the cascade. A negative outcome at any point leads to the immediate rejection of the subwindow (see Fig. 5).

The reason for this strategy is based on the fact that the majority of the subwindows are negative within a single image frame, and it is a rare event for a positive subwindow 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 subwindows as possible, and more computation power will be focused on the more difficult subwindows that passed the scrutiny of the initial stages of the cascade.

Compared with human faces that have comparatively stable image morphology (i.e., the positions of the eyes, the nose, and the mouth on a human face are comparatively stable), hand gestures include multiple shapes, such as the American Sign Language, including 24 different hand poses. To use the Viola and Jones algorithm for hand gesture recognition, we need not only to detect and track different hand postures but to classify them as well.

In our implementation, four hand postures are tested—the “two fingers” posture, the “palm” posture, the “fist” posture, and the “little finger” posture (see Fig. 6). The camera that was used for the video input in our experiment is a low-cost Logitech QuickCam Web camera. This Web camera provides video capture with a maximum resolution of 640×480 up to 15 frames/s. For the experiment, we set the camera parameters at 320×240 with 15 frames/s.

The experiments are implemented in the laboratory with natural fluorescent lighting conditions. To vary the illumination, we installed an extra incandescent light bulb to create a tungsten lighting condition. We collected 480, 412, 400, and 420 positive samples with different scales for the “two fingers” posture, the “palm” posture, the “fist” posture, and the “little finger” posture, respectively. To increase the robustness of the final classifier, we deliberately included a number of positive sam-

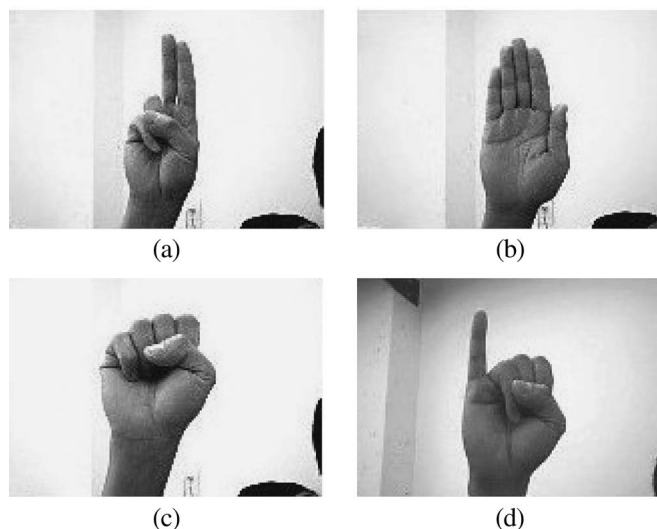


Fig. 6. Gestures tested with the Viola and Jones algorithm.

ples with certain in-plane rotations and out-of-plane rotations. Fig. 7 shows some positive samples that we collected for the “two fingers” posture.

Five hundred random images that do not have the hand posture images are collected as negative samples for the training process. Fig. 8 shows some negative samples that we used. 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 the negative sample images.

After all of the positive and negative samples are ready, we set the required false alarm rate at 1×10^{-6} to terminate the training process, which means that the accuracy of the classifier will meet the requirement when 1 out of 1 million negative subwindows is mistakenly detected as a positive subwindow.

A 15-stage cascade is achieved for the “two fingers” posture when the training process is terminated. During the first training stage, we notice that the positive subwindow detection percentage is 100%, and the negative subwindow elimination percentage is 91%. At the second stage, the negative subwindow elimination percentage reaches 95% while still maintaining the 100% true-positive detection rate, which verifies the advantage of the AdaBoost learning algorithm of eliminating the majority of negative subwindows at the initial stages of the training. When the final required false alarm rate 1×10^{-6} is reached, the true-positive detection rate of the final cascade classifier is 97.5%. For the “palm” posture, a ten-stage cascade is achieved with a true-positive detection rate of 98%. For the “fist” posture, a 15-stage cascade is achieved with a true-positive detection rate of 98%. For the “little finger” posture, a 14-stage cascade is achieved with a true-positive detection rate of 97.1%.

To evaluate the performance of the trained classifiers, 100 test images (which are not used as training samples) for each posture are collected with a similar background but different illumination conditions. Table I shows the performance of the four trained classifiers and the time to process all 100 test images. Fig. 9 shows some of the detection results of the “two fingers” posture.



Fig. 7. Part of the “two fingers” positive samples used in the training.

By analyzing the detection results, we found that most of the positive-false detection results were caused by excessive hand rotations. For the false-positive detection results, the majority of them only happened in very small areas of the images, which can be eliminated by defining a size threshold. The maximum time required for the classifiers to process 100 frames was 3 s in our experiment. The classifiers had in-plane and out-of-plane rotation invariance of $\pm 15^\circ$. The classifiers also showed pretty good robustness against the lighting variance. Background subtraction was used in our implementation to achieve robustness against cluttered backgrounds. A Gaussian filter and image dilation/erosion were used to reduce the noise that is produced by background subtraction. Fig. 10 illustrates the positive effect of these smoothing measures on the resulting hand image. It is noticed that the performance of the Haar-like features improved for the smoothed hand images.

A bank of posture-specific cascade classifiers is assembled for parallel recognition of the hand postures (see Fig. 11). This parallel architecture increases the speed of the hand posture recognition process. A parallel architecture of four cascade classifiers allowed us to obtain real-time recognition of the hand postures with live inputs from the Web camera with 15 frames/s at the resolution of 320×240 . There were neither detectable pause nor latency for the parallel architecture to recognize the hand postures while tracking them. Fig. 12 showed the recognition results.

IV. GESTURE RECOGNITION USING AN SCFG

As a hand gesture is basically an action composed of a sequence of hand postures that are connected by continuous mo-

tions, it makes the idea of describing the hand gesture in terms of a hierarchical composition of simpler hand postures very attractive. To represent the hierarchical structure of hand gestures, the syntactic approach can be applied. With this approach, hand gestures can be specified as building up out of a group of hand postures in various ways of composition, just as phrases are built up by words. The rules governing the composition of hand postures into different hand gestures can be specified by a grammar. The syntactic approach provides the capability of describing a large set of complex hand gestures by using a small set of simple hand postures and grammatical rules.

To describe the structural information about hand gestures, the stochastic context-free grammar (SCFG) is used in our implementation. The SCFG is an extension of the context-free grammar (CFG). Each SCFG is a four-tuple, i.e.,

$$G_S = (V_N, V_T, P_S, S)$$

where V_N and V_T are finite sets of nonterminals and terminals, respectively, $S \in V_N$ is the start symbol, and P_S is a finite set of stochastic production rules, each of which is of the form

$$X \xrightarrow{P} \lambda$$

where $X \in V_N$, $\lambda \in V^+$ (i.e., V_N , V_T , or the combination of them), and P is the probability that is associated with this production rule. The probability P can be also expressed as $P(X \rightarrow \lambda)$, and it satisfies the following:

$$\sum_j P(X \rightarrow \mu_j) = 1$$



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

TABLE I
DETECTION RESULTS OF THE TRAINED CASCADE CLASSIFIERS

Posture Name	Hits	Missed	False	Process Time (seconds)
Two Fingers	100	0	29	3.049
Palm	90	10	0	1.869
Fist	100	0	1	2.829
Little Finger	93	7	2	2.452

where μ_j are all of the strings that are derived from X . In the SCFG, the notion of context-free essentially means that the production rules are conditionally independent [19].

If a string $y \in L(G_S)$ is unambiguous and has a derivation with production rules $r_1, r_2, \dots, r_k \in P_S$, then the probability of y with respect to G_S is given by

$$P(y|G_S) = \prod_{i=1}^k P(r_i).$$

If y is ambiguous and has l different derivation trees with corresponding probabilities $P_1(y|G_S), P_2(y|G_S), \dots, P_l(y|G_S)$, then the probability of y with respect to G_S is given by

$$P(y|G_S) = \sum_{i=1}^l P_i(y|G_S).$$

For more explanations and examples about the SCFG, see [20].

Ivanov and Bobick [21] used the SCFG to recognize activities that are taking place over extended sequences, such as car parking and structured single-stream gestures composed of simple hand trajectories. Minnen *et al.* [22] implemented a system that uses an extended stochastic grammar to recognize a person performing the Towers of Hanoi task from a video sequence by analyzing object interaction events. Moore and Essa [23] presented an experiment of the SCFG to recognize and parsing the card game blackjack.



Fig. 9. Detection result of the trained “two fingers” cascade classifier.

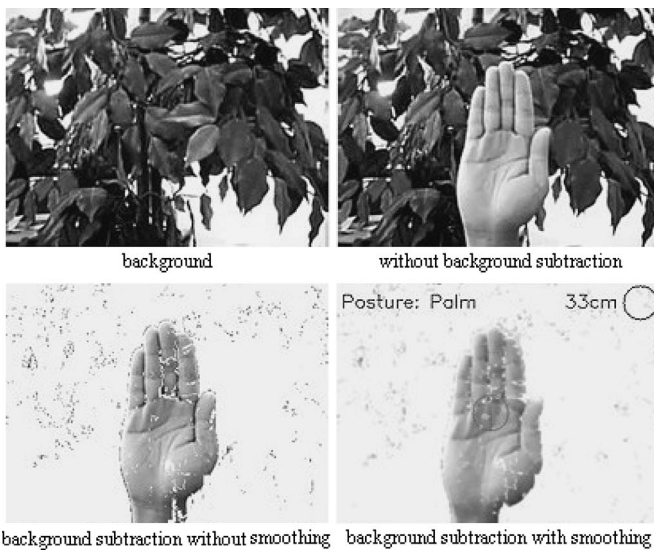


Fig. 10. Effects of background subtraction and smoothing.

The SCFGs extend CFGs in the same way that hidden Markov models (HMMs) extend regular grammars. The relation between the SCFGs and the HMMs is very similar to

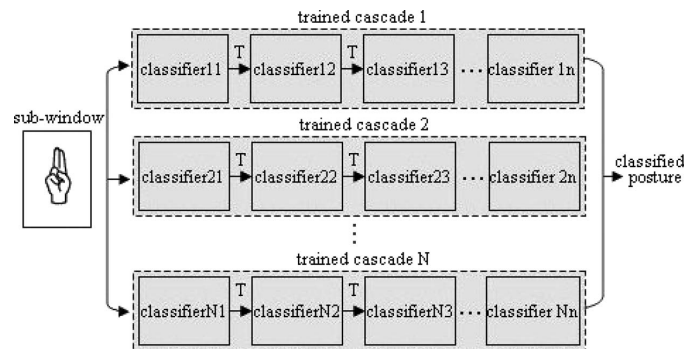


Fig. 11. Parallel cascade architecture for hand posture recognition.

that between the CFGs and the nonprobabilistic finite state machines (FSMs), where the CFGs relax some of the structural limitations imposed by the FSMs, and because of this, the SCFGs have more flexibility than the HMMs [21]. Compared with the nonstochastic CFGs, with the probability attached to each production rule, the SCFGs provide a quantitative basis for ranking and pruning parses as well as for exploiting dependencies in a language model [23]. With the SCFGs, we just need to compute the probability of the pattern belonging to different

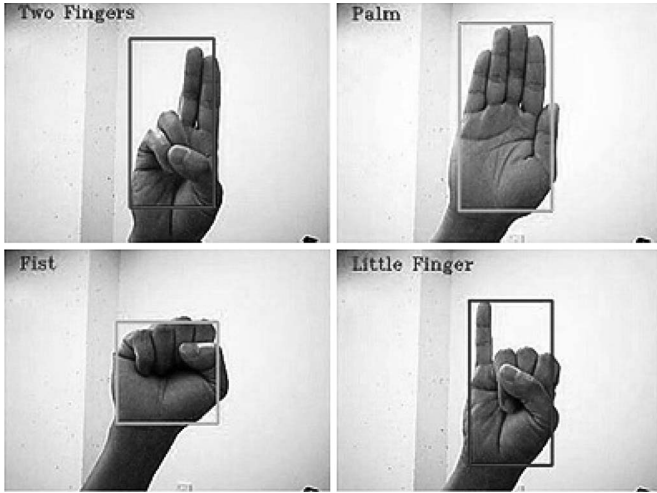


Fig. 12. Recognition results with the parallel cascade architecture.

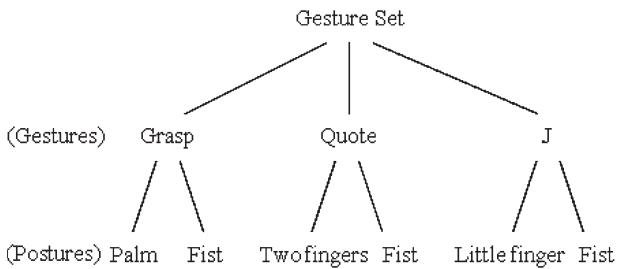


Fig. 13. Gestures that are generated with different postures.

classes (each class is described with its own grammar) and then choose the class with the highest probability value. When an exact parse is not possible, we can still perform an approximate parse and get the probabilities. The probabilistic information allows us to use statistical techniques not only to find the best matches but also to perform robust feature detection that is guided by the grammar’s probability structure.

A simple SCFG is defined in our implementation to generate three different gestures, i.e., “Grasp,” “Quote,” and “J,” with the classified postures. Each gesture is composed of two postures, as illustrated in Fig. 13. The order of the postures included in each gesture is random at this moment. The SCFG that will generate these gestures is defined by $G_G = (V_{NG}, V_{TG}, P_G, S)$, where

$$V_{NG} = \{S\}, \quad V_{TG} = \{t, p, l, f\}$$

and P_G

$$r_1 : S \xrightarrow{\frac{1}{3}} pf, \quad r_2 : S \xrightarrow{\frac{1}{3}} tf, \quad r_3 : S \xrightarrow{\frac{1}{3}} lf.$$

The terminals t, p, l , and f stand for the four postures—“two fingers,” “palm,” “little finger,” and “fist.”

A pipe structure shown in Fig. 14 is implemented to convert the input postures into a sequence of terminal strings.

After a string x is obtained by converting the postures with the pipe structure, we can decide the most likely product rule that can generate this string by computing the probability

$$P(r \Rightarrow x) = D(z_r, x)P(r).$$

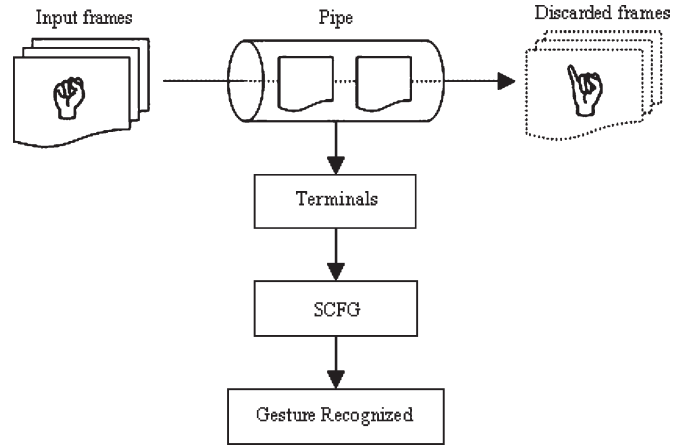


Fig. 14. Pipe structure to convert postures into terminal strings.

$D(z_r, x)$ is the similarity between the input string x and z_r , which is the string derived by the production rule r . $D(z_r, x)$ can be computed according to

$$D(z_r, x) = \frac{\text{Count}(z_r \cap x)}{\text{Count}(z_r) + \text{Count}(x)}.$$

Let us consider that an input string is detected as pf since

$$\begin{aligned} P(r_1 \Rightarrow pf) &= \frac{2}{4} \times \frac{1}{3} = \frac{1}{6} \\ P(r_2 \Rightarrow pf) &= \frac{1}{4} \times \frac{1}{3} = \frac{1}{12} \\ P(r_3 \Rightarrow pf) &= \frac{1}{4} \times \frac{1}{3} = \frac{1}{12}. \end{aligned}$$

According to the highest probability, the gesture represented by this string should be classified as a “Grasp” gesture. Now, consider another input string ll . Since

$$\begin{aligned} P(r_1 \Rightarrow ll) &= \frac{0}{4} \times \frac{1}{3} = 0 \\ P(r_2 \Rightarrow ll) &= \frac{0}{4} \times \frac{1}{3} = 0 \\ P(r_3 \Rightarrow ll) &= \frac{1}{4} \times \frac{1}{3} = \frac{1}{12} \end{aligned}$$

we can say that the gesture represented by string ll is most probably a “J” gesture.

The SCFG can be extended to describe more complex gestures that include more postures as well as the trajectory of the hand movements. The assignment of probability to each production rule can also be used to control the “unwanted” gestures. Very small probability values could be assigned to the “unwanted” gestures such that the resulting SCFG would generate the “wanted” gestures with higher probabilities. For example, we can define P_G as follows:

$$r_1 : S \xrightarrow{\frac{1}{4}} pt, \quad r_2 : S \xrightarrow{\frac{1}{2}} tf, \quad r_3 : S \xrightarrow{\frac{1}{4}} lf$$

where r_2 (corresponding to the “Quote” gesture) is assigned the highest probability. If the input string is detected as tt , since

$$\begin{aligned} P(r_1 \Rightarrow tt) &= \frac{1}{4} \times \frac{1}{4} = \frac{1}{16} \\ P(r_2 \Rightarrow tt) &= \frac{1}{4} \times \frac{1}{2} = \frac{1}{8} \\ P(r_3 \Rightarrow tt) &= \frac{0}{4} \times \frac{1}{4} = 0 \end{aligned}$$

based on the highest probability, the gesture represented by string tt is most probably recognized as a “Quote” gesture. Although the production rule r_1 also has a posture primitive t in its derivation, the associated probability is only $1/4$, which results in a smaller probability to generate the string tt .

V. CONCLUSION

In this paper, we propose 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 the “integral image.” The AdaBoost learning algorithm can greatly speed up performance and construct a strong classifier by combining a sequence of weak classifiers. Based on the trained cascade classifiers, a parallel cascade structure is implemented to classify different hand postures. The experimental results show that this structure can achieve satisfactory real-time performance and high classification accuracy. For high-level hand gesture recognition, we use an SCFG to analyze the syntactic structure based on the detected postures. The postures detected by the lower level are converted into a sequence of terminal strings according to the grammar. Based on the probability that is associated with each production rule, given an input string, the corresponding gesture can be identified by looking for the production rule that has the highest probability of generating it.

The major contributions of this paper are as follows.

- 1) We have achieved real-time performance and accurate recognition for hand postures using Haar-like features and the AdaBoost learning algorithms.
- 2) With the uncertain input of low-level postures, the gesture can be identified by looking for the production rule that has the highest probability of generating the input string.
- 3) The gesture patterns can be controlled by adjusting the probability that is associated with each production rule. Very small probability values could be assigned to the “unwanted” gestures such that the resulting SCFG would generate the gestures we want with higher probabilities.

A system has been developed to demonstrate the proposed approach. The experimental result shows that the system can correctly recognize the hand gestures in real time.

REFERENCES

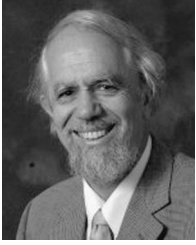
- [1] M. Turk, “Gesture recognition,” in *Handbook of Virtual Environment Technology*. Mahwah, NJ: Lawrence Erlbaum, 2001.
- [2] Y. Wu and T. S. Huang, “Hand modeling analysis and recognition for vision-based human computer interaction,” *IEEE Signal Process. Mag.—Special Issue on Immersive Interactive Technology*, vol. 18, no. 3, pp. 51–60, May 2001.

- [3] H. Zhou and T. S. Huang, “Tracking articulated hand motion with Eigen dynamics analysis,” in *Proc. Int. Conf. Comput. Vis.*, 2003, vol. 2, pp. 1102–1109.
- [4] D. D. Morris and J. M. Reh, “Singularity analysis for articulated object tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1998, pp. 289–296.
- [5] C. Joslin, A. El-Sawah, Q. Chen, and N. Georganas, “Dynamic gesture recognition,” in *Proc. IEEE IMTC*, 2005, pp. 1706–1711.
- [6] Y. Iwai, K. Watanabe, Y. Yagi, and M. Yachida, “Gesture recognition by using colored gloves,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 1996, vol. 1, pp. 76–81.
- [7] R. H. Liang and M. Ouhyoung, “A real-time continuous gesture recognition system for sign language,” in *Proc. 3rd Int. Conf. Autom. Face Gesture Recog.*, 1998, pp. 558–565.
- [8] A. Corradini, “Real-time gesture recognition by means of hybrid recognizers,” in *Proc. Int. Gesture Workshop Gesture Sign Lang. Human-Comput. Interaction*. New York: Springer-Verlag, 2001, vol. 2298, pp. 34–46. Revised paper.
- [9] J. Lin, Y. Wu, and T. S. Huang, “Modeling the constraints of human hand motion,” in *Proc. IEEE Workshop Human Motion*, 2000, pp. 121–126.
- [10] L. Bretzner, I. Laptev, and T. Lindeberg, “Hand gesture recognition using multiscale color features, hierarchical models and particle filtering,” in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recog.*, 2002, pp. 405–410.
- [11] C. W. Ng and S. Ranganath, “Gesture recognition via pose classification,” in *Proc. 15th ICPR*, 2000, vol. 3, pp. 699–704.
- [12] J. Yao and J. R. Cooperstock, “Arm gesture detection in a classroom environment,” in *Proc. IEEE Workshop Appl. Comput. Vis.*, 2002, pp. 153–157.
- [13] G. Bradski, A. Kaehler, and V. Pisarsky, “Learning-based computer vision with Intel’s open source computer vision library,” *Intel Technol. J.*, vol. 9, no. 2, pp. 119–130, May 2005.
- [14] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Boston, MA: PWS-Kent, 1999.
- [15] P. Viola and M. Jones, “Robust real-time object detection,” Cambridge Res. Lab., Cambridge, MA, pp. 1–24, Tech. Rep. CRL2001/01, 2001.
- [16] R. Lienhart and J. Maydt, “An extended set of Haar-like features for rapid object detection,” in *Proc. IEEE Int. Conf. Image Process.*, 2002, vol. 1, pp. 900–903.
- [17] Y. Freund and R. E. Schapire, “A short introduction to boosting,” *J. Jpn. Soc. Artif. Intell.*, vol. 14, no. 5, pp. 771–780, Sep. 1999.
- [18] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2001, vol. 1, pp. 511–518.
- [19] A. Stolcke, “Bayesian learning of probabilistic language models,” M.S. thesis, Univ. California, Berkeley, CA, 1994.
- [20] K. S. Fu, *Syntactic Pattern Recognition and Application*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [21] Y. Ivanov and A. Bobick, “Recognition of visual activities and interactions by stochastic parsing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 852–872, Aug. 2000.
- [22] D. Minnen, I. Essa, and T. Starner, “Expectation grammars: Leveraging high-level expectations for activity recognition,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2003, vol. 2, pp. 626–632.
- [23] D. Moore and I. Essa, “Recognizing multitasked activities using stochastic context-free grammar,” in *Proc. CVPR Workshop Models vs Exemplars Comput. Vis.*, 2001.



Qing Chen received the B.E. degree in 1994 from Jiangnan Petroleum Institute, Hubei, China, the M.E. degree in 1999 from China University of Mining and Technology, Beijing, China, and the M.A.Sc. degree in electrical engineering in 2003 from the University of Ottawa, Ottawa, ON, Canada, where he is currently working toward the Ph.D. degree.

His research interests include visual object detection and tracking, statistical/syntactic pattern recognition, and image processing. His current research topic is focused on vision-based hand gesture recognition in real time.



Nicolas D. Georganas (F'90) received the Dipl.Eng. degree in electrical and mechanical engineering from the National Technical University of Athens, Athens, Greece, in 1966 and the Ph.D. degree (*summa cum laude*) in electrical engineering from the University of Ottawa, Ottawa, ON, Canada, in 1970.

In 2004, he became the Founding Editor-in-Chief of the *ACM Transactions on Multimedia Computing, Communications and Applications*. He is currently a Distinguished University Professor with the School of Information Technology and Engineering, University of Ottawa. He has published over 360 technical papers and is the coauthor of the book *Queueing Networks—Exact Computational Algorithms: A Unified Theory by Decomposition and Aggregation* (Cambridge, MA: MIT Press, 1989). His research interests include multimedia communications, collaborative virtual environments, Web telecollaboration applications, intelligent Internet sensors and appliances, and telehaptics.

Dr. Georganas is a Fellow of the Engineering Institute of Canada, the Canadian Academy of Engineering, and the Royal Society of Canada. He was the recipient of the Killam Prize for Engineering in 2002, the Queens Golden Jubilee Medal in 2003, the Canadian Award in Telecommunications Research in 2006, the first IEEE Canada Computer Medal, the ORION Leadership Award, and an Honorary Doctorate degree from the National Technical University of Athens in 2007. In 2004, he received an Honorary Doctorate degree from the Technical University Darmstadt, Darmstadt, Germany. In 2005, he was recognized as a Pioneer of Computing in Canada by the International Business Machines Corporation Centre of Advanced Studies. In 2007, he was invested as Officer of the Order of Canada: the highest honor for a Canadian.



Emil M. Petriu (M'86–SM'88–F'01) received the Dipl.Eng. and Dr. Eng. degrees from the Polytechnic Institute of Timisoara, Timisoara, Romania, in 1969 and 1978, respectively.

He is currently a Professor and the University Research Chair with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada. During his career, he has published more than 200 technical papers, authored two books, and edited two other books. He is the holder of two patents. His research interests include robot sensing and perception, intelligent sensors, interactive virtual environments, soft computing, and digital integrated circuit testing.

Dr. Petriu is a Fellow of the Canadian Academy of Engineering and of the Engineering Institute of Canada. He is an Associate Editor of the *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT* and a member of the Editorial Board of the *IEEE I&M Magazine*. He is currently the Chair of TC-15 Virtual Systems and the Cochair of TC-28 Instrumentation and Measurement for Robotics and Automation and TC-30 Security and Contraband Detection of the IEEE Instrumentation and Measurement Society. He was a corecipient of the IEEE's Donald G. Fink Prize Paper Award and the recipient of the IEEE Instrumentation and Measurement Society Award in 2003.