Dynamic Sign Language Recognition for Smart Home Interactive Application Using Stochastic Linear Formal Grammar

Muhammad Rizwan Abid, Emil M. Petriu, *Fellow, IEEE*, and Ehsan Amjadian

***Abstract*— This paper presents the state-of-the art dynamic sign language recognition (DSLR) system for smart home interactive applications. Our novel DSLR system comprises two main subsystems: an image processing (IP) module and a stochastic linear formal grammar (SLFG) module. Our IP module enables us to recognize the individual words of the sign language (i.e., a single gesture). In this module, we used the bag-of-features (BOFs) and a local part model approach for bare hand dynamic gesture recognition from a video. We used dense sampling to extract local 3-D multiscale whole-part features. We adopted 3-D histograms of a gradient orientation descriptor to represent features. The *k*-means++ method was applied to cluster the visual words. Dynamic hand gesture classification was conducted using the BOFs and nonlinear support vector machine methods. We used a multiscale local part model to preserve temporal context. The SLFG module analyzes the sentences of the sign language (i.e., sequences of gestures) and determines whether or not they are syntactically valid. Therefore, the DSLR system is not only able to rule out ungrammatical sentences, but it can also make predictions about missing gestures, which, in turn, increases the accuracy of our recognition task. Our IP module alone seals the accuracy of 97% and outperforms any existing bare hand dynamic gesture recognition system. However, by exploiting syntactic pattern recognition, the SLFG module raises this accuracy by 1.65%. This makes the aggregate performance of the DSLR system as accurate as 98.65%.**

***Index Terms*— Dynamic hand gesture, dynamic sign language, formal languages, image processing (IP), local part model, machine learning, stochastic linear formal grammar (SLFG), syntactic pattern recognition.**

I. Introduction

AS THE presence of computers in everyday simple operations of our routines increases, it becomes necessary to make human–computer interaction more intuitive to those who are not used to technical language or who are not immersed in technological advances. For example, sign language is the basic communication method for those who suffer from hearing impairment. The primary component of a sign language is hand gestures. Accordingly, a sign language can be considered as a collection of meaningful and user-friendly hand gestures, movements, and postures. Hand gesture recognition is the most commonly used modality among other communication modalities in human–computer interaction. Dynamic hand gesture communication is a more natural and humanoid mode of communication with computers, as compared with static hand gestures. This is due to the fact that hands in the dynamic mode are allowed to move in any direction, and bend toward any angle in all accessible coordinates. In contrast, static hand gesture communication suffers from a very limited set of possible postures.

Currently, dynamic hand gestures have been adopted by a number of applications, including smart homes, video surveillance, and long-term healthcare environment applications. All of these applications require maximum recognition accuracy and maximum performance against the time and a cluttered background. This paper mainly focuses on a bare hand dynamic gesture recognition application that is based on the robot called Pumpkin, which is developed by the University of Ottawa.

Our dynamic sign language recognition (DSLR) system has some advantages over the previous systems. First, it is composed of two subsystems, namely the image processing (IP) module and the stochastic linear formal grammar (SLFG) module; this modularity per se gives rise to an override mechanism for misses in classification. Therefore, when there is a miss, there is still hope for that instance to be reclassified correctly. Second, older methods (tracking of hand, tracking of fingers, recognition of hand, recognition of fingers, etc) limit the algorithm to that particular object recognition (i.e., that specific part of the body), whereas the bag-of-feature (BOF) approach applied in our IP module eliminates the cumbersome need for tracking and enables us to add any parts of the body for communication without having to modify the algorithm. In addition, it achieves state-of-the-art performance. DSLR is not only good for *static* postures as the previous systems were, but also our choice of the local part model enables DSLR to recognize bare hand *dynamic* hand gesture recognition just as good. Finally, our method also benefits from a linear formal grammar to process a higher level regularity (syntactic constraints) in the sign language, which could not be accessed by the previous systems. This, in turn, adds to the accuracy of the system, making use of what otherwise would have been tagged as just noise.

Manuscript received May 14, 2014; revised August 11, 2014; accepted August 12, 2014. Date of publication September 8, 2014; date of current version February 5, 2015. The Associate Editor coordinating the review process was Dr. Domenico Grimaldi.

M. R. Abid and E. M. Petriu are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: mabid006@uottawa.ca).

E. Amjadian is with the Logic Language and Information Laboratory, Institute of Cognitive Science, Carleton University, Ottawa, ON K1S 5B6, Canada.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2014.2351331

Dense sampling [1] is used to extract local 3-D multiscale, whole part features. It extracts video blocks at regular positions and scales, in both space and time. The 3-D histogram of oriented gradients (3-D HOG) that was proposed in [2] are used to represent features. In addition, there are a number of existing methods for space time feature detection [3]–[8] and description [2], [9]–[12]. Recent evaluation in [1] shows that the combination of dense sampling used to extract local 3-D detectors and 3-D HOG descriptors is ideal to extract and represent features.

Arthur and Vassilvitskii [13] extended the k-means method. They improved both accuracy and running time and proposed the k-means++ method for a cluster technique. They introduced a randomized seeding technique that allows initializing k-means by choosing arbitrary starting centers with very specific probabilities. The technique attempted to minimize the average squared distance between points in the same cluster. This method is fast and accurate, as compared with the k-means. We used k-means++ to cluster our visual words.

Following [14], we used the BOF method and the non linear support vector machine (SVM) for classification. Despite its limitation of only being able to take care of unordered features, BOF became very popular for object classification. Because of its discriminative power, we used this model for gesture recognition of video data. Shi *et al.* [14] proposed a new approach of a 3-D multiscale parts model, which preserved the orders of events. Their model has a coarse primitive level spatio-temporal (ST) feature, as well as word covering and event-content statistics. Conversely, their model has higher resolution overlapping parts that can incorporate temporal relations. By overlapping neighboring subpatches, they can successfully maintain an order of events. Their model produces state-of- the-art results for recognizing dynamic hand gestures. We used this novel approach that is based on fingers as well as hand and arm movements. This idea enabled us to extend hand gestures to include body gestures. This evolution allows for a larger degree of freedom to make meaningful dynamic gestures.

Every language, natural or artificial, has words and sentences. Therefore, any sign language recognition system must be able to recognize both words and sentences. In a dynamic sign language, dynamic gestures are the words. Sequences of these dynamic gestures are sentences of the language. Naturally, however, the ultimate object of recognition is a sentence of the language. We therefore defined a formal language to represent these sentences. We developed formal grammar to accept valid formal language sequences and reject invalid ones. We trained a SLFG that overrides system failure in cases where the gesture is: 1) unrecognizable and 2) recognizable, but rejected by the grammar as invalid. Logically, this goal cannot be achieved with a bare IP module.

This paper is organized as follows. Section II provides a brief overview of related work. Section III describes the IP module architecture overview, which includes a training stage that covers extracting features, use of BOFs, SVM, and k-means++. A testing stage and the recognition of dynamic gestures from our videos are included in this section. This section includes an overview of IP module calibration. Section IV provides an overview of the syntactic pattern extraction performed by the SLFG module. It includes the representation of dynamic hand gesture sequences, a linear formal grammar, and an overview of the training and testing stages of the SLFG. Section V provides our experimental results and compares them with the previous comparable body of work. Section VI provides a conclusion of our work.

II. Related Work

A recent study [15] divided hand detection approaches into two parts: an appearance-based and a model-based approach. In an appearance-based approach, fingertips [16] are detected to enable hand-gesture recognition. The approach uses a neural network-based system that recognizes continuous hand postures from gray level video images. Conversely, in the model-based approach, El-Sawah *et al.* [17] used a histogram for calculating the probability for skin color observation. Several methods of hand detection were proposed, including artificial neural networks/learning-based approaches [16], fuzzy logic, and genetic algorithm-based approaches [18].

Akmeliawati *et al.* [19] proposed the gesture and hand posture-tracking system for New Zealand sign language recognition. They tracked 13 gestures without using any marker. They recognized static postures of a hand and fingers to recognize sign language. They also used markers to attain accurate results for posture recognition. While they attained high-resolution input images for detecting finger postures, the process consequently resulted in a high computational cost.

In [20] and [21], Haar features and AdaBoost are used for performing dynamic classification of the hands. They only recognize static hand postures. They used stochastic, context-free grammar to recognize composite gestures. In addition, they conducted parsing and recognition with a set of production rules.

Dardas and Georganas [22] devised a hand gesture detection and recognition system using the BOF approach, and a multiclass SVM classifier. They built a grammar that generates gesture commands to control applications. Their system is restricted to tracking and recognizing static postures. Their grammar was unable to generate sentences from their static postures. Their system can achieve a satisfactory real-time performance, as well as high classification accuracy under variable conditions. However, full DOF hand pose information is limited by appearance-based methods and may affect the generality of this system.

Varkonyi-Koczy and Tusor [23] conducted modeling of hand postures and gestures. Later, they developed a recognition system of hand gestures to communicate with a smart environment. They used fuzzy neural networks for the recognition of hand gestures. Their system was able to recognize a user's hand gesture thus analyzing the sequence of detected hand postures. They did not recognize dynamic gestures. Consequently, they lacked the ability for humanoid interaction in a smart environment.

Joslin *et al.* [24] introduced a dynamic hand gesture recognition method. Their focus was on tracking fingers and hands to attain information about gestures. They used inverse projection matrices and inverse kinematics to calibrate a hand

model. They applied the hidden Markov model (HMM) to identify and differentiate between gestures. However, this technique was low-speed and was inaccurate because of the high IP cost.

We are motivated by [14], which was originally used for human action recognition. Similarly, some research techniques include a BOF approach that is applied to document analysis. Above all, it has been stated that such approaches prove to function well in controlled settings and in simple backgrounds. Most of the time, these research techniques use global representation and perform background subtraction, object tracking, and skin detection. The main focus of our endeavor is to keep global and local representation information, and to keep track of the order of local events for the dynamic gesture recognition. We chose to use the BOF approach, which will free us from the hand tracking and background subtraction.

III. IP Module: The Dynamic Hand Gesture Recognition System Architecture Overview

To recognize the dynamic hand gestures, our system has been divided into two stages, which comprise training and testing. Initially, we trained our system by extracting features and then clustering them by k-means++. Later, we classified them using a nonlinear SVM. The testing stage used these classifications to recognize dynamic hand gestures.

A. Overview of the Training Stage

First, we generated a database of dynamic gestures for the training stage. We considered six dynamic gestures, which comprised circular, goodbye, rectangular, rotate, triangular, and wait gestures. The circular gesture starts from top, goes to the left side, comes down, moves toward the right, and then back to the top to complete one iteration. This gesture is a combination of hand and arm motions. To complete the goodbye gesture, one begins with one's hand positioned from the top-center and moves toward the left side, and then moves back toward the center and then the right side. The gesture as well involves the arm and the hand. The rectangular gesture movement starts from upper-right side, moves toward the upper-left side, then toward the lower-left, and then to the lower-right, and then moves back up again to its original upper-right position. It also includes both the hand and the arm motion. The rotation gesture is like holding a doorknob with one's fingers and moving along the same axis toward the right and the left directions. This gesture only involves a hand. The triangular gesture begins from the top-center, moves down-left, then toward down-right, and then moves back to its original top-center to its completion. This gesture is based on a combined motion of a hand and an arm. The wait gesture comprises using the wrist to move a hand in a forward and backward motion while the arm remains static. The gestures are shown in Fig. 1.

In total, 16 scenarios have been considered while generating this manual database. These include:

1) straight hand, close to camera, good light, white background;
2) straight hand, far from camera, good light, white background;
3) straight hand, close to camera, bad light, white background;
4) straight hand, far from camera, bad light, white background;
5) angled hand, close to camera, good light, white background;
6) angled hand, far from camera, good light, white background;
7) angled hand, close to camera, bad light, white background;
8) angled hand, far from camera, bad light, white background.

All the above scenarios have been repeated once more with a cluttered background instead of the white background. This raised the number of scenarios to 16. The above mentioned sequences cover all of the possible scenarios of a particular gesture in an environment, which thoroughly increases the robustness of the classification and BOFs model. These sequences are recorded with 40 subjects. For various stages, 30 subjects were considered for the training stage and ten subjects were specified for the testing stage. The researcher captured all of the training clips at 640×480 pixels, and then reduced them to 160×120 pixels. This size reduction increased the feature extraction and classification speed. Most importantly, the size reduction left no effect on the recognition of such features. The training stage model is shown in Fig. 2.

We used the BOF model to extract features from the video sequence. We reduced the size of each video clip by down sampling. A dense sampling approach was used to get 3-D local ST patches of the down-sampled new videos. A multiscale scanning window approach was used to represent the video as a set of features that were computed at different scales and positions. This provided coarse root model features, which contain local global information. From each root model, we extracted high-resolution, overlapping part models. These handsomely incorporated the temporal order information by including the local overlapping part of the dynamic gestures, as shown in Fig. 3. We used a 50% overlapping ratio, as recommended in [14]. Later, a HOG3D [2] descriptor was used to depict the feature from both its root and part models.

We calculated histograms of each root and part model using HOG3D, which is based on a 3-D oriented gradient [25], [26]. HOG3D descriptors can be computed using an integral video method, which is the way to compute spatial-temporal gradient histograms. If we only used local ST features, then the order of dynamic hand gestures would be lost. We overcame this by concatenating histograms of both root and all part models. This maintained the order of dynamic hand gestures. Local ST features have been represented by concatenated histograms. These extracted features are quantized in visual words. Frequencies of these visual words were measured for classification. Word vocabulary was created using dense sampling of dynamic gesture training videos; HOG3D was specifically used for vocabulary representation. We applied the k-means++ [13] method to cluster the features. This approach helped to choose random starting centers with very

Fig. 1. Bare hand dynamic hand gestures.

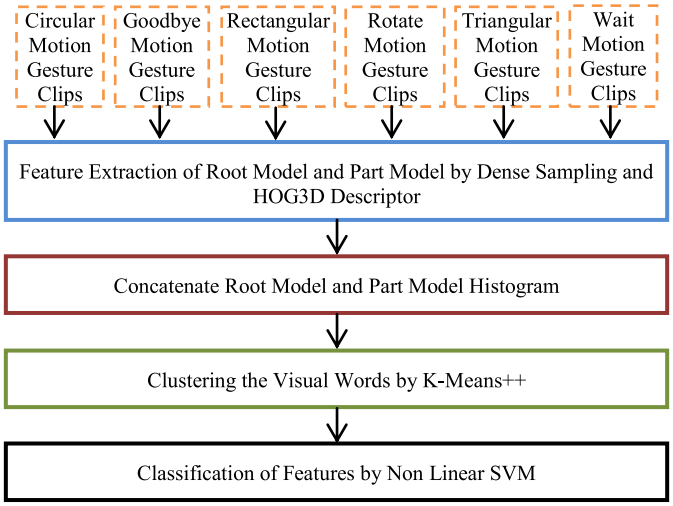

Fig. 2. Overview of training stage.

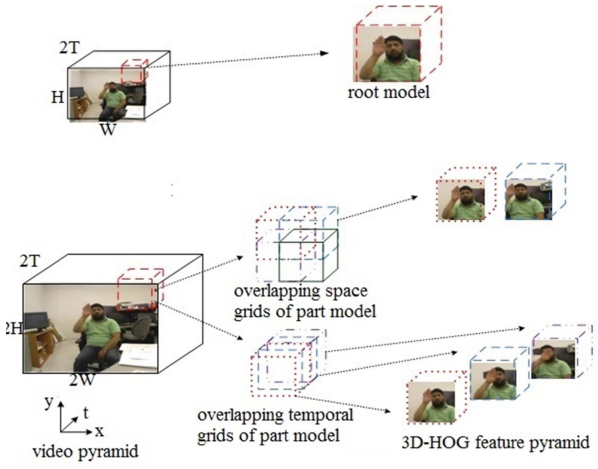

Fig. 3. Feature defined by root model and overlapping grids of part model.

specific probabilities. It proved to be a fast way of sampling. In this process, each feature of the dynamic gesture clips was assigned to the closest Euclidean distance from the vocabulary. Histograms were created to represent the sequence of videos. We used a nonlinear SVM for classification and the library for SVMs [27]. Data scaling was completed on all testing and training data.

B. Overview of the Testing Stage

The testing stage used the same previously mentioned 16 scenarios that were created by the ten subjects.

TABLE I
PERFORMANCE OF GESTURES RECOGNITION WITH SIX GESTURES AND 16 SCENARIOS

Gestures	Correct Recognition out of 160	Incorrect Recognition out of 160	Correct Recognition%
Circular	160	0	100%
Goodbye	151	9	94.4%
Rectangular	150	10	94%
Rotate	152	8	95%
Triangular	159	1	99.4%
Wait	155	5	97%
Total	**927**	**33**	**97%**

We generated another testing database using the same six dynamic gestures. All of the testing clips were captured at 640×480 pixels and then reduced to 160×120 pixels. The parameters values used for sampling and HOG3D were the same as in [14]. The approach was followed with the extraction of the root and part model through the use of dense sampling and the HOG3D descriptor. These extracted features were quantized in visual words. Our local part model integrated the temporal order information by including local overlapping events. The integration provided more discriminative power for dynamic gesture recognition. The approach used the concatenated root model and part model histograms to maintain the order of the local events for dynamic hand gesture recognition.

We used k-means++ to conduct clustering of the visual words from the visual vocabulary. This clustering technique chooses randomly located centroids (points in space that represent the center of the cluster), and it assigns every key point to the nearest centroid. Later, centroids are moved to the average positions of all assigned keypoints and the assignments are redone. This process is completed when the assignments stop changing. We used a randomized seeding technique that allows initializing k-means by choosing arbitrary starting centers with very specific probabilities. The technique attempted to minimize the average squared distance between points in the same cluster.

Later, the approach applies the BOF and nonlinear SVM approach for classification. It assigns different values to the codebook (visual vocabulary). We conducted experiments with different codebook size values, including 1000, 2000, 3000, 4000, and 5000 words. We chose to use increments of 1000s to demonstrate significant results; minor changes like 1002 would not significantly influence the result. The ideal value for a codebook size is 4000 words. We may achieve a better percentage of recognition by increasing the codebook size. However, increasing the size after 4000 will increase the computational cost without resulting in a significant improvement. The ratio of increments in computational cost and precision of results is ideal for increments between 1000 and 4000.

These calculated increments enable a 97% aggregated result of gesture recognition with six gestures and 16 scenarios. Some gestures result in 100% recognition, as mentioned in Table I. The approach shows the performance of dynamic hand gesture recognition by the SVM classifier. Each gesture has been thoroughly and properly tested against scale, rotation, illumination, and numerous backgrounds.

TABLE II
DYNAMIC GESTURES WITH NOISE AND THEIR RESULTS

Dynamic Gestures with Noise	Checked Against	Results
Half circle complete bye	Bye	Pass
Half circle complete bye	Circle	Fail
Complete bye half circle	Bye	Pass
Complete bye half circle	Circle	Fail

C. IP Module Calibration

We know now that our IP module is able to categorize dynamic gestures with good accuracy. For instance, it can recognize that a triangular gesture is not a circular one and vice versa. However, we were also interested to know if it could reject dynamic gestures that did not belong to any of the trained categories. Furthermore, we wanted to make sure our system did not wrongly categorize some gesture with only partial characteristics of a gesture class. For example, half a circle is not a circle. Consequently, we conducted a calibration test as follows.

We also checked our algorithm accuracy by deliberately creating noise in simulation videos.

We created a new database by recording videos of a human hand with combinations of one iteration of one gesture and a half iteration of another gesture and checked our system recognition accuracy against both gestures. For example, we recorded one video with one iteration of a bye gesture and a half iteration of a circle gesture and checked it against both bye and circle gestures. Results are shown in Table II. These videos were made by considering the same parameters discussed in [28]. The gestures were captured to .avi video format using H.263 video codec. Each video displayed gestures at a resolution of 640×480 pixels, which were then reduced to 160×120 pixels at 30 frames/s. Each dynamic gesture was recorded multiple times with the same parameters as mentioned in [28].

We checked all combinations of dynamic gestures. However, due to lack of space, samples of video results are shown in Table II. These results led us to the successful implementation of the grammar and formal language of our dynamic sign language; below we mention why in more detail.

Prior to the test reported in Table II, we had also created a testset of dynamic hand gestures with solely one iteration of each gesture to see if only one iteration of a dynamic gesture was recognizable by our system. All of our results were successful.

Aside from testing for noisy environments, the tests conducted in this section were important for two practical reasons. Firstly, we wanted the convenience of command cancellation, like for instance, if a human initiated a command, but canceled it halfway through by not completing the gesture (e.g., because

TABLE III
DYNAMIC HAND GESTURES AND THE CORRESPONDING COMMANDS AND SYMBOLS OF THE FORMAL LANGUAGE

Dynamic Hand Gestures	Commands	Symbols
Wait	Wake up	s
Circular	Open both hands	a
Goodbye	Bye	b
Rectangular	Close both hands	c
Rotate	Raise both arms	d
Triangular	Lower both arms	e

he figures it is an invalid command). Secondly, it is vital for the IP module to have accurate information to train its system on. Therefore, we cannot allow canceled cases/noise to be fed in.

IV. SLFG MODULE: SYNTACTIC PATTERN EXTRACTION

During the phase of any type of interaction, whether artificial or natural, the information flow might be distorted, which will lead to a communication failure. This phenomenon is often conceived to be the result of unprecedented noise in the channel, and is therefore inevitable. Hence, for any pattern recognition system that is aimed to be designated online, a noise handling strategy proves to be vital. In addition to the existing noise in the channel, the flow of information from source to target might be intentionally discontinued or disrupted due to various motives that include but are not bound to the following.

1) The sender decides whether the missing bits are inferable and therefore avoids explicitly sending them. Here, the information flow is distorted intentionally.
2) The sender is unable to keep the intended information flow to complete the message. Here, the information flow is distorted unintentionally.

No matter how accurately a pattern recognition system works, it cannot classify information that it is not exposed to. To overcome this limitation, we used the following strategy.

1) We first abstracted from the process of recognizing dynamic signs to recognize sequences of dynamic signs. Ultimately, we defined a formal language to represent these sequences.
2) We developed formal grammar to accept valid sequences of the formal language and to reject invalid ones.
3) We trained a stochastic linear grammar that overrides system failure in cases where the gesture is unrecognizable, missing, or is present and recognizable, but is rejected by the grammar as invalid.

We mapped our dynamic hand gestures onto the corresponding symbols as shown in Table III, which, in turn, can form sentences. It is worth mentioning that there is a distinction between the gesture itself and the command assigned to it. The commands and their corresponding actions are as follows.

1) Wake up: robot raises his neck, forehead and says hi.
2) Open both hands: the robot angles at approximately 180° between thumbs and fingers.
3) Bye: the robot lowers his neck, forehead and says bye. This keyword puts the robot in a state of rest.
4) Close both hands: the robot angles approximately at 0° between thumbs and fingers.
5) Raise both arms: the robot angles at approximately 90° between forearms and arms.
6) Lower both arms: the robot angles at approximately 180° between forearms and arms.

Next, we will see what we mean by a formal language, formal grammar and how they help to construct a successful interaction.

A. G_1: Linear Formal Grammar

A chain of commands may be invalid not only because one of the elements in the chain is invalid, but also because a command is not in the right place, or in the right context. For example, it is meaningless for Pumpkin to be turned on right after it is turned ON. Be that as it may, two or more consecutive turn ON commands are invalid. This notion can be fully captured by our formal grammar called G_1.

The concepts of formal language and grammar are inseparable. A formal language is a set of sentences that can be generated by a formal grammar [{Tan, 2010 #10} 29], which is defined as $G = (V_N, V_T, P, S)$, where V_N, V_T, and P are finite sets, nonempty and:

1) $V_N \cap V_T = 0$;
2) $P \subseteq V^+ \times V^*$;
3) $S \in V_N$.

In the above-mentioned system, V_N denotes the finite set of nonterminal symbols. We can regard them as *nodes*. V_T denotes the finite set of terminal symbols that can be regarded as *leaves;* these are the actual strings or commands. Finally, $V = V_N \cup V_T$. S does not denote a sentence of the system; rather, it is the start symbol from where the machine starts the derivation. P is the finite set of the production rules from which the system generates the sentences, or by which the system accepts the sentences. V^* is the set of all finite strings that are an element of V, and V^+ is V^* minus the null-string represented by λ. The above system exclusively generates/accepts the valid concatenation of the strings of a language, which are called the sentences of that language.

Now, we can present our linear grammar G_1

$$G_1 = (V_N, V_{T,} P, S).$$

Our terminal vocabulary V_T is composed of the strings that our dynamic hand gestures were previously mapped onto

$$V_T = \{s, a, b, c, d, e\}.$$

Our nonterminal vocabulary V_N is as follows:

$$V_N = \{S, A, B, C, D, E\}.$$

G_1 production rules P are in the following Table IV.

Our linear grammar G_1 distinguishes between invalid sequences of commands and valid ones; only then our system goes on and corrects the invalid ones. Alternatively put, our system knows what an invalid sequence of commands is, and then it tries to guess the correct sequence intended

TABLE IV
PRODUCTION RULES OF OUR FORMAL LANGUAGE

Production Rules			
S→sA	A→aE	C→cA	D→dE
S→sB	A→aD	C→cB	E→e
S→sC	B→b	C→cD	E→eA
S→sD	B→bA	C→cE	E→eB
S→sE	B→bC	D→d	E→eC
A→a	B→bD	D→dA	E→eD
A→aB	B→bE	D→dB	
A→aC	C→c	D→dC	

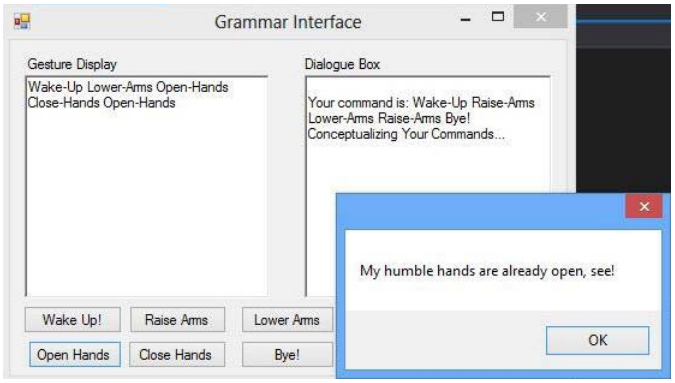

Fig. 4. GI to train the SLFG module.

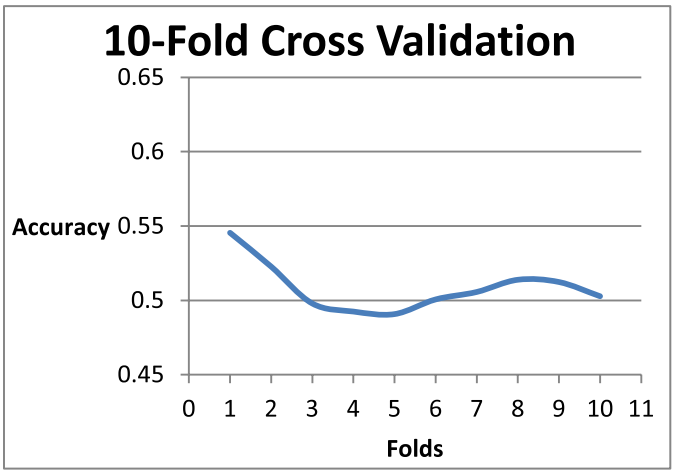

Fig. 5. Ten-fold CV graph.

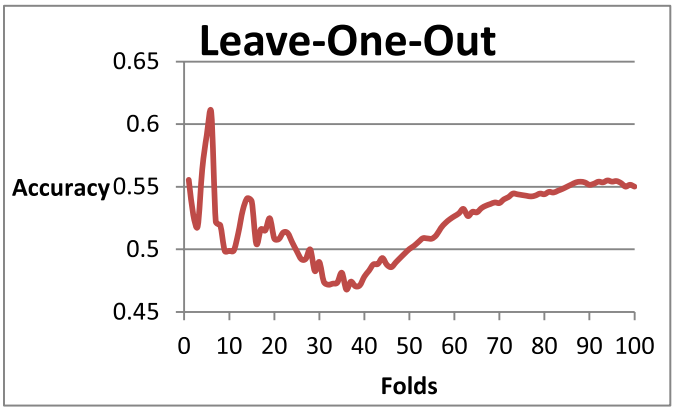

Fig. 6. LOO graph.

by the sender before performing it. This helps to avoid a machine halt. In the theory of formal languages and automata, linear grammar is the most restricted one; type three, as first mentioned in [30]. Hence, they are the most efficient in terms of processing and computational cost. They are equivalent to finite state machines. G_1 captures linear context surrounding each gesture, or what validly precedes or follows each gesture.

B. Training and Testing an SLFG

We trained the SLFG module by (1), which is the general method to train SLFGs, as follows:

$$1 \cdot p(C \rightarrow cD) = \frac{N(C \rightarrow cD)}{N(C)}.$$

We developed a software named Grammar Interface (GI) to train the SLFG module. GI guarantees that only valid sequences of gestures find their way into the training set. Fig. 4 displays GIs interface when it rejects to accept a gesture into its dialog box. It rejects the gesture since the resulting input gesture sequence is not allowed by the rules of G_1. Using Table II, we can translate the sequence of gestures into the formal language. The input gestures shown under the Gesture Display box in Fig. 4 are: Wake-Up Lower-Arms Open-Hands Close-Hands Open-Hands. In addition, when the next gesture is introduced as Open-Hands, the rejection message will pop up. The reason is that the sequence *s e a c* ***a a*** cannot be produced by the rules of G_1 (previously shown in Table IV), and therefore, is not a valid sequence of the language.

The dataset is made of 100 sentences of our formal language with an average length of ten words. The longest sentence in the dataset is composed of 28 words, and the shortest is two words long.

SLFG computes the likelihood of a sequence among all of the possible sequences pertaining to a missing/invalid/unrecognized gesture.

We used two methods to test the accuracy of the SLFG module: tenfold cross validation (tenfold CV) and leave-one-out (LOO).

1) Tenfold Cross Validation: In the tenfold CV method, the data which consist of 100 sentences were divided into ten equal and nonoverlapping folds. One of the folds is held out each time, and the other nine folds are used to train the SLFG. Then the unseen held-out fold is used to test the system. This process is repeated for every fold (i.e., 10 times). The fold boundary parameters were set by the number of sentences. That is, for each training and testing pair, the data set was split into 90 sentences for training and 10 remaining separate sentences for testing. Fig. 5 presents the results of the tenfold CV.

2) Leave-One-Out: In the LOO method, each time one sentence of our data set, which consists of 100 sentences

TABLE V
COMPARISON OF OUR EXPERIMENT WITH PREVIOUS APPROACHES

Approaches	No of Gestures	Frame Resolution	Light	Background	Number of Parameters per Gesture	Number of Scenarios per Gesture	Aggregate Number of Cases	Recognition Accuracy %
Dynamic Bayesian Network[31]	12	24fps 320*240	No change	White	0	1	12	90
Motion Trajectories and Key Frames[32]	4	Not mention	No change	Black	0	1	4	96
SURF Tracking[33]	26	8-16fps 176*144	No change	Less clutter	1	2	52	84.6
Hidden Markov Model[34]	18	20fps 640*480	No change	Black	0	1	18	96.67
DTW and Multi class Probability[35]	12	Not mention	No change	Clutter	1	2	24	96.85
Our Approach(without grammar)	**6**	**30fps 160*120**	**varied**	**Clutter**	**4**	**16**	**96**	**97**
Our Approach (with grammar)	**6**	**30fps 160*120**	**varied**	**Clutter**	**4**	**16**	**96**	**98.65**

in total, is held out and the SLFG module is trained by the other 99 sentences. The system performance is then tested against the held-out unseen sentence. This cycle is iterated 100 times until the performance is tested against each individual sentence. The aggregate accuracy of the LOO validation is presented in Fig. 6.

V. RESULTS AND COMPARISON

In our experiments, we chose parameter settings to make it computationally tractable, mainly by limiting the vector size of visual words. The optimal parameter settings are: codebook size $= 4000$; minimal patch size $= 12, 6$; total sampling scales $8 \times 8 \times 3$; number of histogram cells $2 \times 2 \times 2$; polyhedron type dodecahedron (12); and number of parts per root model $2 \times 2 \times 2$. The dimension for root model is $2 \times 2 \times 2 \times 12 = 96$. The vector size of a feature is $96 \times (1(\text{root}) + 8(\text{parts})) = 864$. We conducted experiments with different values, but we concluded that parameter values suggested in [14] are the best combination.

We compare the results of our experiments with comparable previous works. Shiravandi *et al.* [31] used a dynamic Bayesian network method for dynamic hand gesture recognition. They considered 12 gestures for recognition. They achieved an average accuracy of 90%. Wenjun *et al.* [32] proposed an approach based on motion trajectories of hands and hand shapes of the key frames. The hand gesture of the key frame is considered as a static hand gesture. The feature of hand shape is represented with a Fourier descriptor and is recognized by the neural network. The combined method of the motion trajectories and the key frame is presented to recognize the dynamic hand gesture from unaided video sequences. They consider four dynamic hand gestures for experiment. Their average recognition accuracy is 96%. Bao *et al.* [33] did dynamic hand gesture recognition based on speeded up robust features (SURFs) tracking. The main characteristic is that the dominant movement direction of matched SURF points in adjacent frames is used to help describe a hand trajectory without detecting and segmenting the hand region. They consider 26 alphabetical hand gestures, and their average recognition accuracy rate achieved was 84.6%. Yang *et al.* [34] proposed the HMM for hand gesture recognition. They consider 18 gestures for recognition. Their recognition rate is 96.67%. Pisharady and Saerbeck [35] used dynamic time wrapping and multiclass probability estimates to detect and recognize hand gestures. They used kinect to get skeletal data. They claimed 96.85% recognition accuracy with 12 gestures. The above mentioned results and our result comparison are given in Table V. As shown in the table, our experiment achieved the highest accuracy although we had the highest number of aggregate cases, parameters, and scenarios.

VI. CONCLUSION

The Human Computer Interaction system presented in this paper applied a novel method to recognize dynamic hand gestures, and achieved the state-of-the-art performance with a recognition rate of 98.65%. The system is composed of two main modules. The IP module recognizes the dynamic hand gestures (our visual words) as accurately as 97% by

applying a novel technique for the task, and our SLFG module uses an SLFG to raise it by another 1.65%. It manages to obtain this increase by predicting the intended gesture from pure noise. The SLFG module also checks if the commands are valid by the use of a linear formal grammar.

It is worth noting that the SLFG module can be mounted on any other dynamic hand gesture recognition system for any inherently sequential task.

The IP module uses a new strategy to recognize dynamic bare hand gestures from a video stream that was never applied to this domain. We used a HOG3D descriptor and a dense sampling approach for feature extraction from the root and part model. We then concatenated histograms of the root and part model. This approach helps to maintain the sequence of events. We performed vector quantization using the k-means++ clustering technique, which produced visual words from a visual vocabulary. We then applied BOF and non linear SVM techniques to achieve classification. Our experiments showed that this combination of techniques achieves satisfactory performance under a combination of variable scale, orientation, background, and illumination conditions.

This research contributes to human–robot communication by making dynamic gestures available as a trustworthy modality; in healthcare domain, it benefits hearing impaired patients who use dynamic signs as the preferred modality. It also helps patients with speech disorders communicate with smart devices through dynamic gestures, which is a natural mode of human communication. Humans communicate via multiple modalities, one modality facilitates recognition of the other; hence, the present work may be applied to and be combined with speech processing technology to facilitate speech recognition that, in turn, improves human–robot interaction; and therefore, it is a reasonable direction for future research. Another direction may improve the system to achieve real-time performance. Expanding the DSLR system to include other parts of the body for communication may be yet another direction this paper leads to.

REFERENCES

[1] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. BMVC*, 2009, pp. 124.1–124.11.

[2] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2008, pp. 995–1004.

[3] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. 2nd Joint IEEE Int. Workshop VS-PETS*, Oct. 2005, pp. 65–72.

[4] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proc. ICCV*, Oct. 2007, pp. 1–8.

[5] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proc. ICCV*, Oct. 2003, pp. 432–439.

[6] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal salient points for visual recognition of human actions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 3, pp. 710–719, Jun. 2006.

[7] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. 10th ECCV*, Oct. 2008, pp. 650–663.

[8] K.-Y. K. Wong and R. Cipolla, "Extracting spatiotemporal interest points using global information," in *Proc. ICCV*, Oct. 2007, pp. 1–8.

[9] I. Laptev and T. Lindeberg, "Local descriptors for spatio-temporal recognition," in *Proc. 1st Int. Workshop Spatial Coherence Vis. Motion Anal.*, 2004, pp. 91–103.

[10] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. CVPR*, Jun. 2008, pp. 1–8.

[11] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. ACM Int. Conf. Multimedia*, 2007, pp. 357–360.

[12] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. ECCV*, 2008, pp. 650–663.

[13] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 1027–1035.

[14] F. Shi, E. M. Petriu, and A. Cordeiro, "Human action recognition from local part model," in *Proc. IEEE Int. Workshop Haptic Audio Vis. Environ. Games (HAVE)*, Oct. 2011, pp. 35–38.

[15] A. Chaudhary, J. L. Raheja, K. Das, and S. Raheja, "A survey on hand gesture recognition in context of soft computing," in *Proc. CCSIT*, vol. 133. Jan. 2011, pp. 46–55.

[16] C. Nolker and H. Ritter, "Visual recognition of continuous hand postures," *IEEE Trans. Neural Netw.*, vol. 13, no. 4, pp. 983–994, Jul. 2002.

[17] A. El-Sawah, N. D. Georganas, and E. M. Petriu, "A prototype for 3-D hand tracking and posture estimation," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 8, pp. 1627–1636, Aug. 2008.

[18] C. Hu, Q. Yu, Y. Li, and S. Ma, "Extraction of parametric human model for posture recognition using genetic algorithm," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, Mar. 2000, pp. 518–523.

[19] R. Akmeliawati *et al.*, "Towards real-time sign language analysis via markerless gesture tracking," in *Proc. Instrum. Meas. Technol. Conf.*, May 2009, pp. 1200–1204.

[20] Q. Chen, N. D. Georganas, and E. M. Petriu, "Real-time vision-based hand gesture recognition using Haar-like features," in *Proc. Instrum. Meas. Technol. Conf.*, May 2007, pp. 1–6.

[21] Q. Chen, N. D. Georganas, and E. M. Petriu, "Hand gesture recognition using Haar-like features and a stochastic context-free grammar," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 8, pp. 1562–1571, Aug. 2008.

[22] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 11, pp. 3592–3607, Nov. 2011.

[23] A. R. Varkonyi-Koczy and B. Tusor, "Human–computer interaction for smart environment applications using fuzzy hand posture and gesture models," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 5, pp. 1505–1514, May 2011.

[24] C. Joslin, A. El-Sawah, Q. Chen, and N. Georganas, "Dynamic gesture recognition," in *Proc. IEEE IMTC*, May 2005, pp. 1706–1711.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *Proc. IJCV*, Nov. 2004, pp. 91–110.

[26] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[28] M. R. Abid, L. B. S. Melo, and E. M. Petriu, "Dynamic sign language and voice recognition for smart home interactive application," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, May 2013, pp. 139–144.

[29] W. J. M. Levelt, *An Introduction to the Theory of Formal Languages and Automata*. Amsterdam, The Netherlands: John Benjamins Pub., 2008.

[30] N. Comsky, "On certain formal properties of grammars," *Inf. Control*, vol. 2, no. 2, pp. 137–167, 1959.

[31] S. Shiravandi, M. Rahmati, and F. Mahmoudi, "Hand gestures recognition using dynamic Bayesian networks," in *Proc. 3rd Joint Conf. AI Robot. 5th RoboCup Iran Open Int. Symp. (RIOS)*, Apr. 2013, pp. 1–6.

[32] T. Wenjun, W. Chengdong, Z. Shuying, and J. Li, "Dynamic hand gesture recognition using motion trajectories and key frames," in *Proc. 2nd Int. Conf. Adv. Comput. Control (ICACC)*, Mar. 2010, pp. 163–167.

[33] J. Bao, A. Song, Y. Guo, and H. Tang, "Dynamic hand gesture recognition based on SURF tracking," in *Proc. Int. Conf. Elect. Inf. Control Eng. (ICEICE)*, Apr. 2011, pp. 338–341.

[34] Z. Yang, Y. Li, W. Chen, and Y. Zheng, "Dynamic hand gesture recognition using hidden Markov models," in *Proc. 7th Int. Conf. Comput. Sci. Edu. (ICCSE)*, Jul. 2012, pp. 360–365.

[35] P. K. Pisharady and M. Saerbeck, "Robust gesture detection and recognition using dynamic time warping and multi-class probability estimates," in *Proc. IEEE Symp. Comput. Intell. Multimedia, Signal Vis. Process. (CIMSIVP)*, Apr. 2013, pp. 30–36.

Muhammad Rizwan Abid received the bachelor's degree in computer science from the University of the Punjab, Lahore, Pakistan, and the master's degree in computer science from the University of Ottawa, Ottawa, ON, Canada, in 2008, where he is currently pursuing the Ph.D. degree in computer science.

He joined Amtex (Pvt) Ltd., Faisalabad, Pakistan, as an Application Developer. He was involved in software engineering research on the design and implementation of a UML profile for goal-oriented modeling. He is an Oracle Certified Application Developer and Oracle Certified Internet Application Developer. His current research is focused on dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar. His current research interests include multimedia communication, human–computer communication, human–robot and inter-robot communication, automata theory, and formal languages.

Emil M. Petriu (M'86–SM'88–F'01) is currently a Professor and University Research Chair with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. He has authored over 480 papers and book chapters, authored two books, edited two other books, and holds two patents. His current research interests include biology-inspired robot sensing, soft computing, human–computer interaction, and system-on-a-chip design.

Dr. Petriu is a fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada. He was a co-recipient of the 2003 IEEE's Donald G. Fink Prize Paper Award, the 2003 IEEE Instrumentation and Measurement Society Technical Award, and the 2009 IEEE Instrumentation and Measurement Society Distinguished Service Award.

Ehsan Amjadian received the M.Sc. degree in linguistics-computational linguistics from Allameh Tabatabai University, Tehran, Iran. He is currently pursuing the Ph.D. degree in cognitive science with Carleton University, Ottawa, ON, Canada.

He is involved in research on information extraction, in particular, coreference resolutions systems, and automatic terminology extraction. His current research interests include statistical natural language processing, statistical and syntactic pattern recognition, and automata theory and formal languages.

Mr. Amjadian was a member of the Department of Computational Linguistics with the Linguistics Society of Iran, and the Association for Computational Linguistics.