Visual Recognition of a Dynamic Arm Gesture Language for Human-Robot and Inter-Robot Communication

Muhammad Rizwan Abid

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Doctorate of Philosophy in Computer Science

Under the auspices of the Ottawa-Carleton Institute for Computer Science

University of Ottawa
Ottawa, Ontario, Canada
July 2015

Abstract

This thesis presents a novel Dynamic Gesture Language Recognition (DGLR) system for human-robot and inter-robot communication.

We developed and implemented an experimental setup consisting of a humanoid robot/android able to recognize and execute in real time all the arm gestures of the Dynamic Gesture Language (DGL) in similar way as humans do.

Our DGLR system comprises two main subsystems: an image processing (IP) module and a linguistic recognition system (LRS) module. The IP module enables recognizing individual DGL gestures. In this module, we use the bag-of-features (BOFs) and a local part model approach for dynamic gesture recognition from images. Dynamic gesture classification is conducted using the BOFs and nonlinear support-vector-machine (SVM) methods. The multiscale local part model preserves the temporal context.

The IP module was tested using two databases, one consisting of images of a human performing a series of dynamic arm gestures under different environmental conditions and a second database consisting of images of an android performing the same series of arm gestures.

The linguistic recognition system (LRS) module uses a novel formal grammar approach to accept DGL-wise valid sequences of dynamic gestures and reject invalid ones. LRS consists of two subsystems: one using a Linear Formal Grammar (LFG) to derive the valid sequence of dynamic gestures and another using a Stochastic Linear Formal Grammar (SLFG) to occasionally recover gestures that were unrecognized by the IP module. Experimental results have shown that the DGLR system had a slightly better overall performance when recognizing gestures made by a human subject (98.92% recognition rate) than those made by the android (97.42% recognition rate).

Acknowledgment

I am first of all thankful to Allah, who has provided me this opportunity of doing Doctorate of Philosophy in Computer Science degree at University of Ottawa.

I would like to express my deep gratitude and special thanks to my supervisors, Dr. Emil M. Petriu and Late Dr. Nicolas D. Georganas, who have supported me with invaluable suggestions, knowledge and positive encouragements during my thesis work. I have gained enormously from their ideas, technical insights and profound thinking. They gave me opportunity to shape all my future.

I also wish to offer my special thanks to Dr. Feng Shi, Ehsan Amjadian and my lab fellows for our great discussions and comments that provided me with the inspiration to complete this thesis.

Finally, I want to thanks my parents, family members and especially my wife for her endless support throughout my studies and my mother who supported and encouraged me from past to the present. Great thanks for my lovely son Abdullah and daughter Horein for being patients while I'm busy and away from them.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
2D	Two Dimensional
3D	Three Dimensional
AdaBoost	Adaptive Boost
ANN	Artificial Neural Network
ASL	American Sign Language
BOF	Bag-Of-Feature
BOW	Bag Of Words
CAD	Computer Aided Design
CAMShift	Continuously Adaptive Mean Shift
CFG	Context-Free Grammar
DFSM	Deterministic Finite State Machine
DGL	Dynamic Gesture Language
DGL	Dynamic Gesture Language
DGLR	Dynamic Gesture Language Recognition
DNA	DeoxyriboNucleic Acid
DOF	Degree Of Freedom
DOG	Difference of Gaussian
DTW	Dynamic Time Wrapping
GI	Grammar Interface
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
HOF	Histograms of Optical Flow
HOG	Histograms of Oriented Gradients
HOG3D	Histograms of Oriented 3D spatio-temporal Gradients
ICA	Independent Component Analysis
IP	Image Processing

JSL	Japanese Sign Language
KLT Tracker	Kanade-Lucas-Tomasi Tracker
LEDs	Light Emitting Diodes
LFG	Linear Formal Grammar
LIBSVM	Library for Support Vector Machine
LRS	Linguistic Recognition System
LRS	Linguistic Recognition System
MRI	Magnetic Resonance Imaging
OAO	One Against One
PCA	Principal Component Analysis
RBF	Radial Basis Function
SIFT	Scale Invariant Feature Transform
SLFG	Stochastic Linear Formal Grammar
ST	Spatio-Temporal
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
Tenfold CV	Tenfold Cross Validation
UML	Unified Modelling Language

Chapter 1. Introduction

1.1. Concepts and Context

Arm gestures are a collection of movements of the hand, forearm, elbow and upper arm. They are distinct from the static hand signs made using special hand finger configurations as for instance those used in the American Sign Language (ASL). This concept will be used throughout this thesis.

Arm gestures represent a powerful natural communication modality between humans, providing a major information transfer channel in our everyday life, transcending language barriers. Hand signs and dynamic arm gestures are an easy to use non-verbal communication modality of humans with other humans and even with some animals. For example, sign languages have already been used extensively among speech or hear-disabled people. Even people who can speak and hear also use many kinds of arm gestures and hand signs to help their communication in audio noisy environments or too far away for hearing applications (e.g. on board of ships or aircraft carriers).

More recently, arm gesturing also became a major communication modality for human-computer interaction (HCI). Dynamic gesture communication is a very natural and human-like mode of communication with computers, which complements the static hand signs. As a result of arm being able to move in any direction and to bend to almost any angle in all coordinates, dynamic arm gesturing adds a new dynamic dimension complementing the static hand gestures are limited to significantly less meaning set of hand postures.

Visual arm movement recognition is done in both spatial and temporal domains. It requires high accuracy in terms of recognition and time, as well as level of perfection against a cluttered background, variable light condition and variable distance.

Visual recognition of dynamic arm gestures has recently been adopted by a number of applications, like smart homes, video surveillance, human-robot communication in smart homes, healthcare and eldercare applications.

1.2. Motivation

Recent years have produced a remarkable growth in the development of new interfaces for human-computer interaction (HCI). These techniques are offering more convenient *human-centric communication modalities* that make human-computer communication more natural and intuitive, as illustrated in Figure 1, [1] [2] [3].

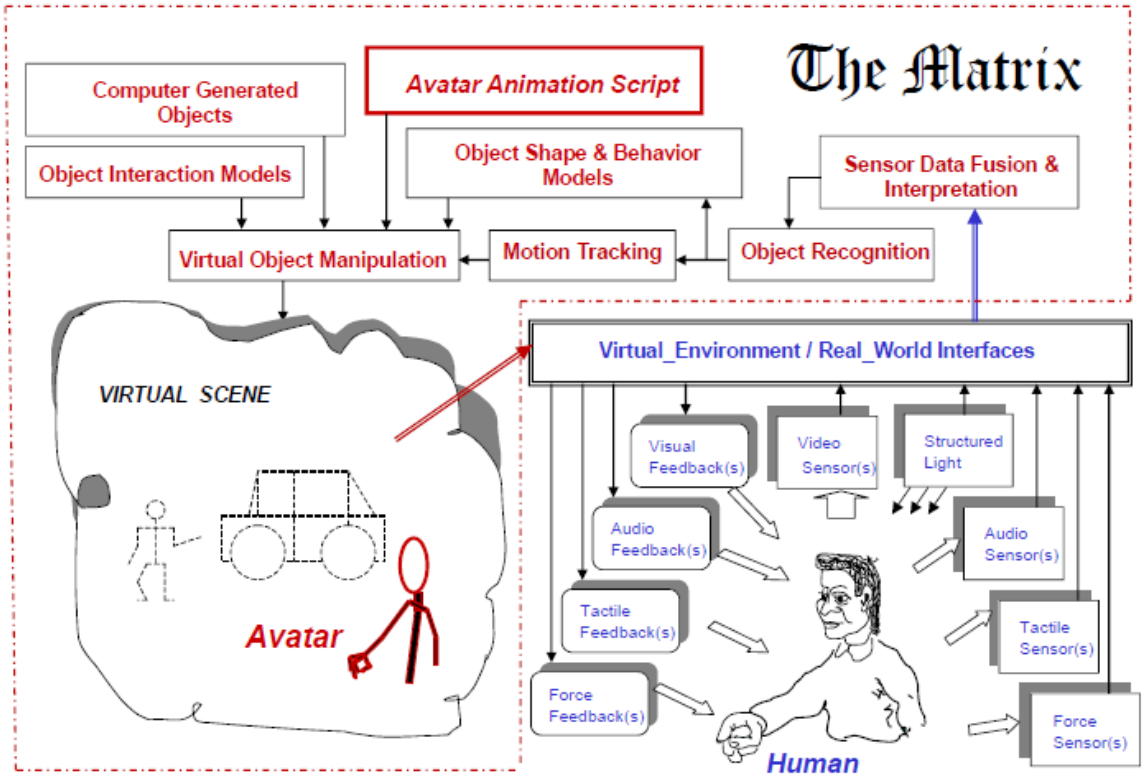


Figure 1 Human-centric HCIs for virtual environment applications [1]

In particular, dynamic arm gesture interfaces are beginning to attract the attention of developers as they provide a powerful addition to the static hand signs, such as the already well established American Sign Language (ASL) and Japanese Sign Language (JSL), in many fields of activities where humans and intelligent humanoid robots (a.k.a androids), are working together, such as in manufacturing, industrial maintenance operations, disaster-management interventions, healthcare, or humanitarian missions [4] [5] [6] [7]. These intelligent androids should be more responsive to human behavior and

also be able to learn from experience and by observing how humans behave, communicate and perform different tasks.

A new generation of humanoid robots is currently under development in order to help older adults "age in place," improve their quality of-life and reduce healthcare expenditures by allowing these people to live independently in their own household for a longer time, fulfilling their desire for longer autonomy and independence.

As arm gesturing is one of the basic natural, instinctive communication modality for humans, it is naturally to expect that it would also be adopted by these androids. Gesture communication ability would complement other bi-directional robot-human communication modalities such as the verbal and face-expression recognition, allowing androids to more naturally mingle and interact with humans and other humanoid robots (androids) while they are all cursorily interacting in normal household activities, as illustrated in Figure 2.

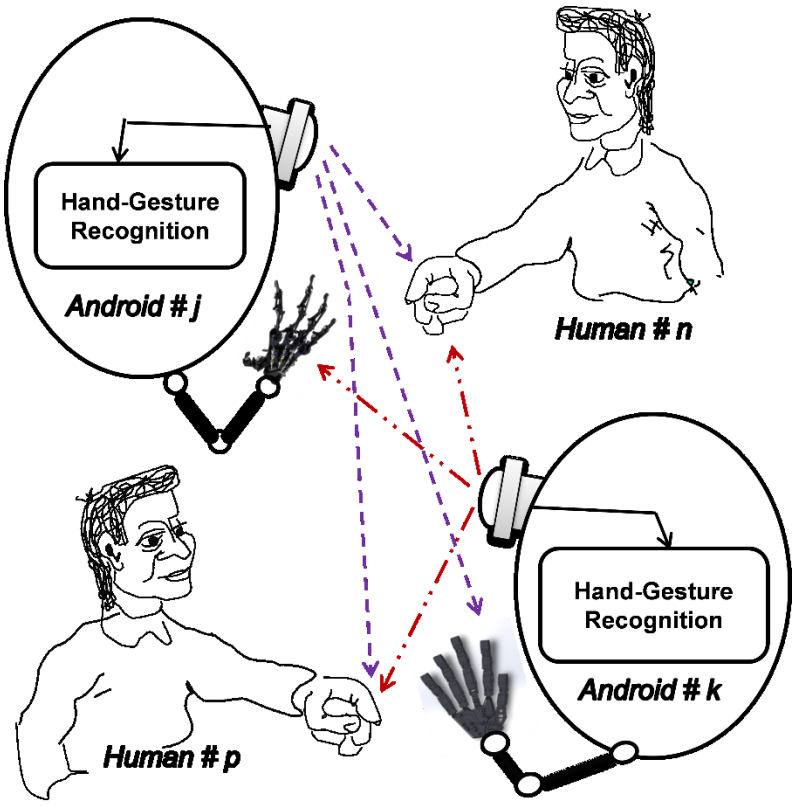


Figure 2 Visual recognition of gestures made by androids communicating with humans and other androids.

1.3. Objective and Approach

The main objective of this thesis is the development of an efficient *dynamic gesture language* (DGL) that both androids (humanoid robot) and humans can comfortably use. In order to do this we will need to solve the following two major problems:

(i) Find a set of meaningful dynamic arm gestures that robots can adequately perform in a human-like fashion;

(ii) Develop robust computer vision methods to allow robots to recognize the dynamic arm gestures made by other robots as well as by humans, and to understand the meaning of the Dynamic Gesture Language (DGL) expressions defined by sequences of valid recognized gestures.

A dynamic arm gesture communication system between robots and humans should be human friendly and capable of integration with other human-centric HCIs such as the already well established static hand sign languages (e.g. ASL), or the body-language and face-expression recognition interfaces.

Dynamic arm gesture recognition is a challenging task in computer vision. It has to be computer-efficient, should be accurate and robust in order to consistently recognize the correct dynamic gestures. Dynamic gesture recognition is even more challenging a task when the background is cluttered or when occlusion occurs. In this thesis, we are mainly concerned with the development of a robust dynamic arm gesture recognition system that can adequately perform under visually challenging settings.

These challenging settings bring a higher degree of uncertainty in most classification systems. A number of gestures organized classes in an n-dimensional feature space can be used as a basis to recognize new gestures if these new inputs are within a certain range of similarity. Figure 3 shows the two dimensional (2D) case, where the each white instance is classified according to its closest cluster. In order to improve the recognition performance, a dynamic gesture language recognition (DGLR) system must be able to categorize efficiently noisy gestures, represented by the yellow dot in the Figure 3.

There are a number of computer vision recognition techniques used by researchers but recently bag-of-feature (BOF) became very popular for action recognition due to its simplicity and good performance. Regardless of its limitations due to its ability to take care

of only unordered features, it became very popular for many object classification tasks. Because of its discriminative power we use the BOF method and the nonlinear "support vector machine (SVM)" classification technique for the dynamic gesture recognition from video data.

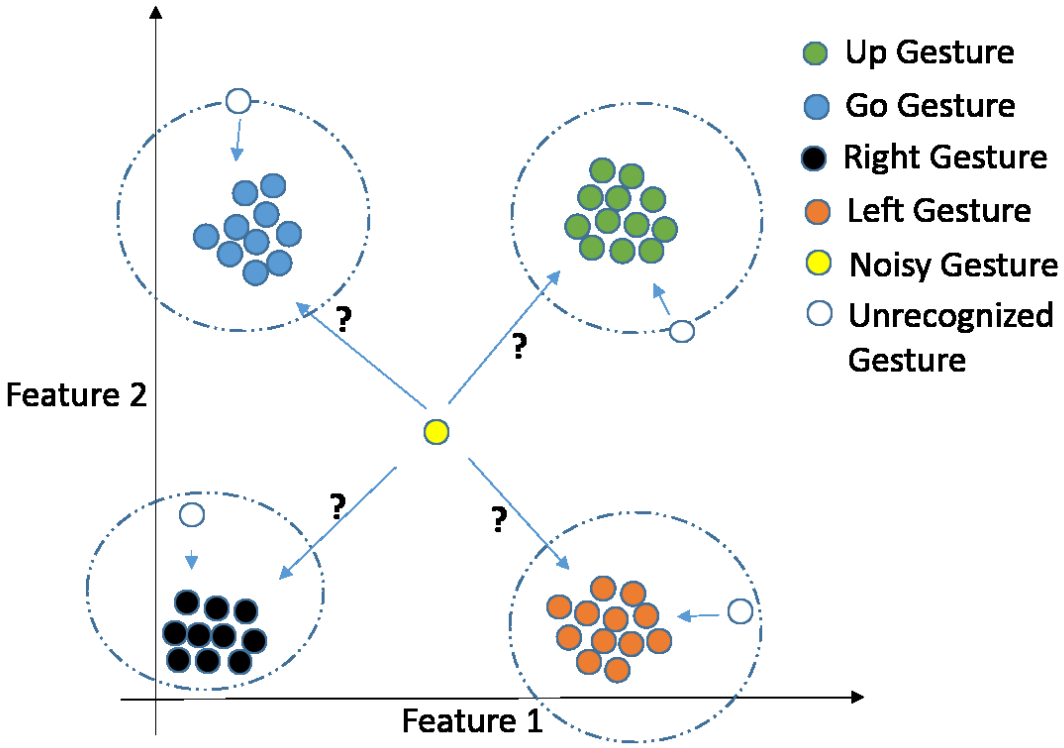


Figure 3 Gesture recognition in feature space.

We will extend Shi et al.'s [8] work, which was originally used for human action recognition. Similarly some research approaches include a *bag-of-feature* (BOF) approach that is applied to document analysis. It has been found that this approach proves to function well in controlled settings and simple backgrounds. Most of the time, it uses global representation and performs background subtraction, object tracking and skin detection. The main focus is to keep global and local representation information, and to keep track of the order of local events for dynamic gesture recognition. We chose the BOF approach as it will contribute to the robustness of our system by eliminating the need to perform arm tracking and background subtraction tasks. This technique also allows for more of a

freedom of making meaningful dynamic gestures which includes combination of hand, fingers, wrist, elbow, forearm and upper arm.

As our proposed dynamic gesture language (DGL) has a relatively small set of gestural expressions, the grammar that generates these expressions could be a simple regular grammar. This will allow us to employ a grammatical syntactic approach for the recovery of uncertain individual dynamic gestures which were not fully recognized by BOF but only guessed/estimated by the nearest neighbor method. The guessed value/symbol of the uncertainly recovered gesture is validated through a short-term memory parsing algorithm.

A stochastic linear formal grammar will be developed for Dynamic Gesture Language Recognition (DGLR). This will increase the classification accuracy by adding the notion of probabilities and short-term memory to tackle the presence of noisy gestures in DGL sequences. Figure 4 illustrates a sequence of gestures starting with "Go", transitioning to "Right", followed by "Left". At this point the system finds a noisy gesture that cannot be recognized properly by the image processing module. The stochastic linear formal grammar is used to calculate the probabilities P1 and P2 based on the previous gestures and on the corpus generated by production rules of a linear formal grammar.

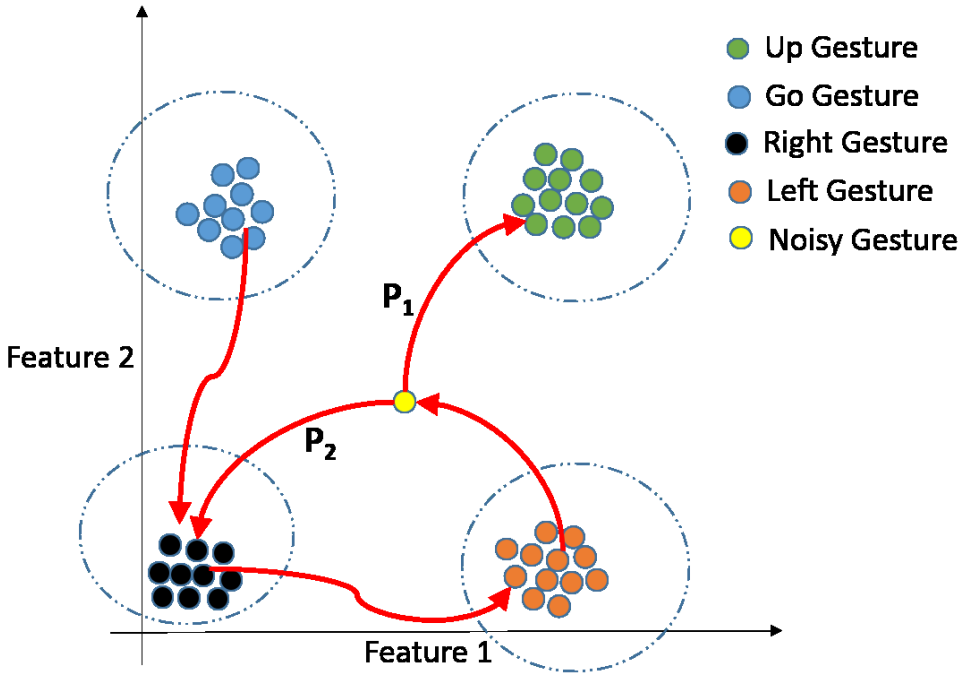


Figure 4 Gesture recognition using precedents.

While common sign recognition systems use only the visual information in the data, our Dynamic Gesture Language Recognition system will use both the primary visual information and the DGL syntactic information so it will make more informed decisions while performing the classification task.

1.4. Thesis Contributions

This thesis presents a novel Dynamic Gesture Language Recognition (DGLR) system to facilitate the communication among heterogeneous groups including androids as well as humans. The DGLR system comprises two main subsystems: an image processing (IP) module and a linguistic recognition system (LRS) module.

The following summarizes the contributions of the thesis:

- Dynamic arm gesture recognition using bag of features (BOF) and local part model approach. We proposed this method because previous methods (tracking of hand, tracking of fingers, recognition of hand, recognition of fingers, etc.) limit the algorithm to specific parts of the body. In contrast, our approach eliminates the need to track specific body parts and has ability to expand our work from the recognition of one-arm dynamic gestures to both-arms dynamic gestures. This approach allowed us to achieve an overall accuracy of 98.92% in the case of human gestures and 97.42% in the case of android gestures. These are state of the art results as compare to other researcher's recognition results presented in —Table 8.
- Modular open architecture for dynamic gesture language recognition (DGLR) system including an image processing (IP) module and a linguistic recognition system (LRS) module. This modularity makes the modules separately upgradable, so the improvement of one doesn't need changing the other, while increasing the overall performance of the system.
- A novel linguistic recognition system (LRS) module that analyzes the sequences of gestures of the gesture language and determines whether or not they are syntactically valid. We developed a novel formal grammar approach to accept valid sequences of the formal language and reject invalid

ones. We consider the dynamic gestures as symbols (words) and construct sentences (sequences of gestures) by the use of a formal grammar that enables the system to understand the dynamic gesture language.

- We developed a consistent testing procedure allowing to measure the gesture recognition accuracy for each of the DGL gestures under different environmental conditions. For each DGL gesture we have 24 scenarios as mentioned in chapter 3. So in total we have 12x24=288 scenarios e.g. for tilted arm, full front arm, vertical arm, good light, bad light, with white background, with cluttered background etc.
- We successfully developed and implemented an experimental set up consisting of a human-sized android able to execute in real time all the DGL dynamic arm gestures in similar way as humans do.

1.5. Limitations

There are some limitations necessary to be mentioned:

- All our work is not based on real time scenarios but on recorded videos. We trained and tested our image processing (IP) module with recorded video clips.
- Our linguistic recognition system (LRS) module may not be as efficient for the more complex sequence.

1.6. Publications Arising from this Thesis

The following publications have arisen from the work presented in this thesis:

- **M. R. Abid**, E.M. Petriu, E. Amjadian, "**Dynamic Sign Language Recognition for Smart Home Interactive Application Using Stochastic Linear Formal Grammar**," IEEE Trans. Instrum. Meas., vol.64, no.3, pp.596-605, March 2015. [9]

- **M. R. Abid**, P.E. Meszaros, R.F. da Silva, E.M. Petriu, "**Dynamic Hand Gesture Recognition for Human-Robot and Inter-Robot Communication**," Proc. CIVEMSA 2014 - IEEE Int. Conf. on Computational Intelligence and Virtual Environments for Meas. Systems and Applications, pp. 12-17, Ottawa, ON, Canada, May 2014. [10]

- **M. R. Abid**, L.B. Santiago Melo, E.M. Petriu, "**Dynamic Sign Language and Voice Recognition for Smart Home Interactive Application**," Proc. MeMeA2013, 8th IEEE Int. Symp. on Medical Measurement and Applications, pp. 139-144, Ottawa, ON, Canada, May 2013. [11]

- **M. R. Abid**, F. Shi, E.M. Petriu, "**Dynamic Hand Gesture Recognition from Bag-of-Features and Local Part Model**," Proc. HAVE 2012 - IEEE Int. Symp. Haptic Audio Visual Environments and Games, 78 – 82, Munich, Germany, Oct. 2012. [12]

- Q. Chen, F. Malric, Y. Zhang, **M. Abid**, A. Cordeiro, E.M. Petriu, N.D. Georganas, "**Interacting with Digital Signage Using Hand Gestures**", Proc. ICIAR 2009, Int. Conf. Image Analysis and Recognition, (M. Kamel and A. Campilho - Eds), Lecture Notes in Computer Science Vol. LNCS 5627, pp. 347-358, Springer, Berlin/Heidelberg, 2009. [13]

- **M. R. Abid**, "**Dynamic Hand Gesture Recognition for Human-Computer and Inter-Robot Interaction**" poster in 8th Edition of Engineering and Computer Science Graduate Poster Competition, Univ. of Ottawa, Canada, March 31, 2015, **Scored Third Position**.

1.7. Thesis Outline

The rest of the thesis is as follows: Chapter 2 presents the background work of gesture recognition approaches. Our main focus is on vision based gesture recognition approaches. Chapter 3 presents first module of the Dynamic Gesture Language Recognition (DGLR) system and also presents our method of local part model and bag-of-feature for dynamic arm gesture recognition. An action recognition framework will be illustrated. The system introduced in this chapter leads to the construction of the Image Processing (IP) module. Chapter 4 discusses the inter-robot communication architecture. Here we discussed about dynamic arm gesture recognition for androids. This will show the robustness of our algorithm and will prove our algorithm accuracy and efficiency in presence of noise. Chapter 5 explains dynamic gesture language recognition using formal grammars. It introduces the Linguistic Recognition System (LRS) module and describes how it improves the accuracy of the overall system (i.e. DGLR). The LRS is the second main module of the DGLR system proposed. Chapter 6 concludes the thesis and puts forth the future work.

Chapter 2. Background

Because of the human arm's high degree of freedom (DOF), arm gesture recognition is a challenging problem. To better understand the complexity of the arm gestures there are two important aspects that should be considered [14] [15].

Hand Sign: a static hand pose with no movement involved and no change in its location.

Arm Gesture: a sequence of arm signs connected by continuous palm, fingers, wrist, forearm, elbow or upper arm movements.

Aiming is to produce a more natural human-like way of communication, our thesis will explore new intelligent computer vision methods for arm gesture recognition.

As already mentioned a *hand sign* is defined as a static hand pose, as for instance, making a certain finger arrangement and holding it is considered to be a hand posture. An *arm gesture* is defined as a dynamic arm movement involving one's fingers, palm, wrist, forearm, elbow or upper arm. An example of an arm gesture is someone waving good bye. Because of its manifoldness applications and its ability to efficiently interact with other machines through HCI, arm gesture recognition systems have recently received more attention. A comprehensive literature review of this field would be an intimidating challenge. Because of the significant amount of publications about the topic, this chapter will be a review of the most representative developments that are pertinent to my thesis research topics.

Most of the research done up to date concentrate on the hand sign recognition using different HCIs, while relatively little was done and published about the dynamic arm gesture recognition for both humans and androids communication. This is the reason for the background information provided in this chapter deals mostly with hand sign systems.

The first research attempts to develop hand sign recognition systems for HCIs have used glove-based devices [16] [17] [18] [19]. However, a glove-based interface requires the user to wear a device that connects to a computer with cumbersome cables. This arrangement lacks ease and the natural, humanoid way of communicating [16]. Figure 5 shows the gloves that have been used for HCI.

Figure 5 Gloves used for HCI

In order to collect data from these instrumented gloves [20], computer input devices are physically attached to a user's hand. These gloves report data values for finger movements; the amount of reported data values depends on the type of glove worn. Depending on the glove type, trackers are attached to the back of the hand or the upper wrist. They collect data on the position and orientation of the hand in three Dimensional (3D) space. These sensor devices are very hectic; in order to collect data, a user must be wearing the gloves. This device restricts the user's freedom of movement. These limitations have spurred research about the use of computer vision to track human movements and to recognize static and dynamic hand signs.

2.1. Vision Based Hand Sign Recognition

Vision based hand sign recognition techniques are a more natural human-like way of communicating. This method of recognition/tracking provides a researcher with more DOF; it also does not require the use of gloves. Vision based recognition

systems requires application specific image processing algorithms, programming, and machine learning.

In this technique, a camera is used to capture the data, which includes hand position static postures. Researcher will typically use one camera to capture an image, but may also use more than two cameras, depending on the scenario. Multiple cameras [21] [22] increase the algorithmic complexity of simultaneously dealing with multiple data sources. This mode of recognition can experience a number of challenges, like robust recognition, occlusion handling, variable light conditions and ensuring an appropriate distance between the camera and an object. Researchers find various ways to deal with these challenges. For the better hand-to-camera visibility, Stuman et al. use LEDs (light emitting diodes) [20] on certain points of the hand. Some introduce color gloves [21] [23] [24] for better visibility. While this option lacks the finger bending and moving, a solution was found by replacing solid coloured gloves with colored rings around the finger joints [25]. More recent research has categorized vision based hand sign recognition into the following approaches:

- 3D Hand Model Based Approaches
- Appearance Based Approaches
- Machine Learning Algorithm Based Approaches
- Rule Based Approaches
- Syntactic Approaches
- Local Feature Approaches

2.1.1 3D Model Based Approaches

Three dimensional (3D) hand model based approaches [26] [27] [28] [29] [30] rely on 3D kinematic hand models that have considerable DOF's, and that try to estimate the hand parameters by comparing the input images and the possible 2D appearance projected by the 3D hand model. Such an approach is ideal for realistic interactions in virtual environments. The 3D hand model based approach provides a rich description that permits a wide class of hand signs. However, since 3D hand models are articulated deformable objects with many DOF’s, a large image

database is required to deal with the entire characteristic shapes under several views. The 3D hand model based approaches search a region in the high dimensional space by using three steps. Initially, the algorithm proposes hypothesis parameters based on prior knowledge, like previously recovered hand configuration and hand dynamics. The algorithm then uses the hypothesis parameters to animate the 3D hand model, it projects the model into a two dimensional (2D) image and then compares it with the input image. The algorithm continues to vary the hypothesis parameters until the projected image matches the observation image and successfully classifies the hand sign [31].

This approach has several disadvantages that have prevented it from real-world use. At each frame, the initial parameters must be close to the solution, or the approach is liable to find a suboptimal solution (i.e. local minima). Secondly, the fitting process is also sensitive to noise (e.g. lens aberrations, sensor noise) in the imaging process. Finally, the approach cannot handle the inevitable self-occlusion of the hand.

2.1.2 Appearance Based Approaches

Appearance based approaches extract the features of the image to model the visual appearance of the hand. The next step is to compare these features with the extracted features from the video frames using a pattern classification module [32] [33]. Figure 6 shows the model of appearance based techniques.

Appearance based approaches centre on the direct registration of hand signs with 2D image features. Skin color is an important image feature to detect a human hand and to recognize postures. However, it is difficult to distinguish the hand with other objects having the same color, such as an arm and a face. Stenger in [34], proposed a method to search for skin coloured regions in the image. This proposed method is highly sensitive to lighting conditions and is based on the pre assumption that an image has no other skin-like objects. This approach has a higher probability to accidentally mix a hand with a face or an arm. Bretzner et al. [35], used scale space color features to recognize hand signs. This method also shows best results only when there is no other skin coloured object present in the image.

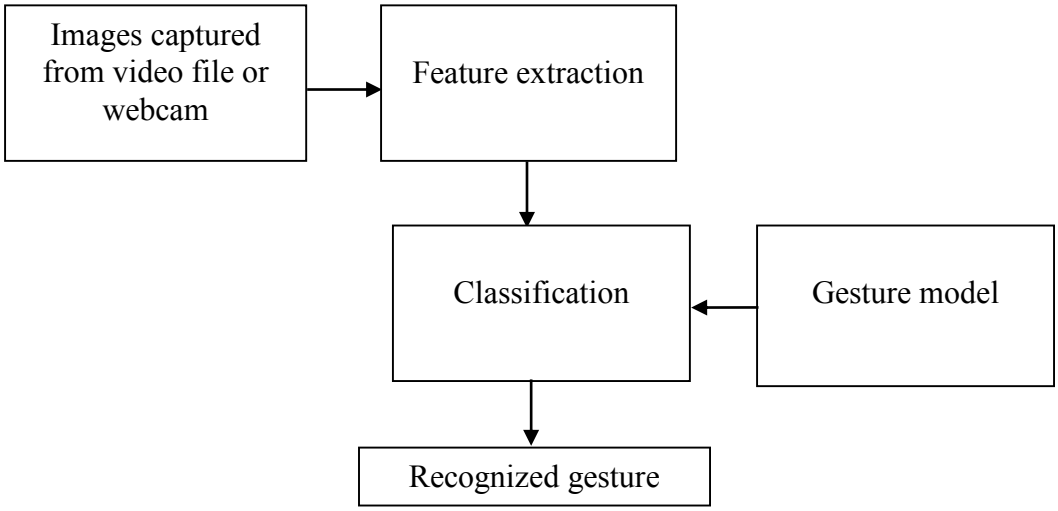


Figure 6 Appearance based approach processing stages

In [36], Ng and Ranganath proposed a global shape descriptor to represent different hand shapes. They used Zernik moments and Fourier descriptors as global shape descriptors. The computational cost of most shape descriptors is considerably high, because they are pixel based. Shape based methods also result in higher noise during image segmentation. A cluttered background decreases the performance of the shape based method. Bowden et al. in [37], proposed an approach that provides a high classification rate on minimal training data for sign language recognition. They divided the classification process in two stages. The first classification stage extracts a high level description of hand shape and motion. To achieve this, they used sing linguistics and action descriptions at a conceptual level, which was easy for humans to understand. In the second stage, they modeled the temporal transitions of individual signs using a bank of Markov chains combined with Independent Component Analysis (ICA).

Oka et al. in [38], proposed a method for locating fingertip positions in image frames and measuring fingertips trajectories across image frames. They developed an augmented desk interface, which depends on accurate, real time hand and fingertip tracking to integrate between real objects and associated digital information. Their method can track multiple fingertips by using an infrared

camera under a complex background and variable light conditions. They did not use a color marker or any invasive device for this tracking. Zhang et al. in [39], proposed a vision based interface named a "visual panel," as is shown in Figure 7. The visual panel is based on an ordinary piece of paper and a fingertip pointer as an intuitive input device. The system precisely tracks the panel and the fingertip pointer.

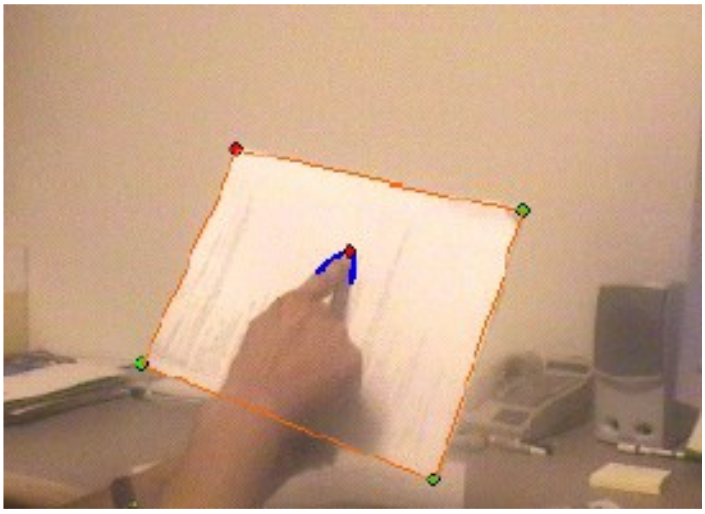


Figure 7 The visual panel system [39]

After the system recognizes “click and drag” actions performed by hand, it can fulfill respective tasks associated with those actions, such as controlling a remote large display. In [40], Malik et al. proposed a plane-based augmented reality system. They made it possible to interact with virtual objects with a fingertip-based recognition system. Figure 8 shows that their system depends on a number of fingertips detected in the image. Single fingertip detection can be assigned with a pointing sign, while multiple fingertip detection can be assigned with a selection sign. Their proposed method completed the detection of fingers through background subtraction and by scanning the binary image for pixels of full intensity.

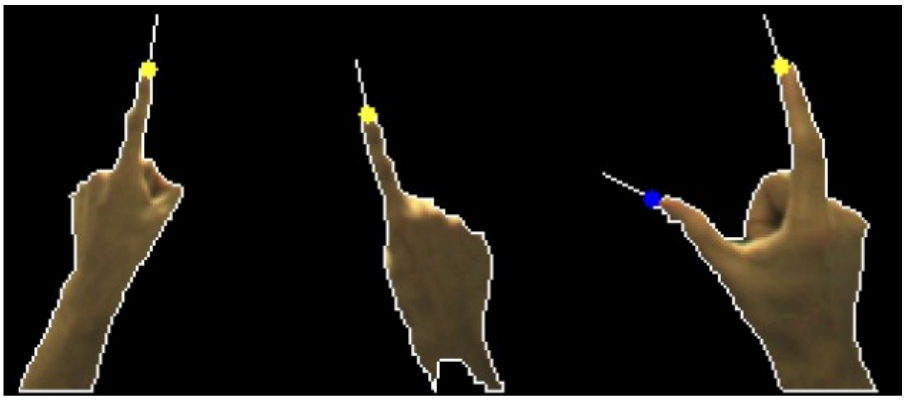


Figure 8 Hand sign recognition based on fingertips [40]

Yuan et al. in [41], proposed a 2D hand tracking approach. This method extracts trajectories of unambiguous hand locations. They use a novel feature based approach, the foundation of which includes motion residue to detect hand bounded by squares, as shown in Figure 9. They also combined a skin detection approach in color video. A temporal filter employs the Viterbi algorithm to identify consistent hand trajectories. Their experiments show robustness of hand tracking in video sequences of several hundred frames.

Figure 9 Hand tracking using motion residue and hand color [41]

Another algorithm, Mean Shift, has been used by number of researchers to characterize an object of interest. It is a simple, iterative procedure that shifts the center of the region of interest to the center of mass (i.e. the average of the data points), as shown in Figure 10. The statistical distributions characterize the object of interest. Comaniciu et al. in [42], presented an approach for non-rigid objects tracked in real time by using mean shift iterations. He successfully found the most probable target position in the current frame. Another researcher, Bradski, in [43], proposed a modified form of a mean shift algorithm, which he named Continuously Adaptive Mean Shift (CAMShift). This approach is specifically designed to perform efficient head and face tracking in a perceptual user interface. This algorithm uses a color probability image frame to find the center and size of the object. A histogram model specific color is used to create probability. The search window is moved and resized by the tracker until its center converges with the center of mass. Some results of the CAMShift based hand tracking are shown in Figure 11. This algorithm performance is compromised when needed to classify different hand postures that might share a similar center of mass.

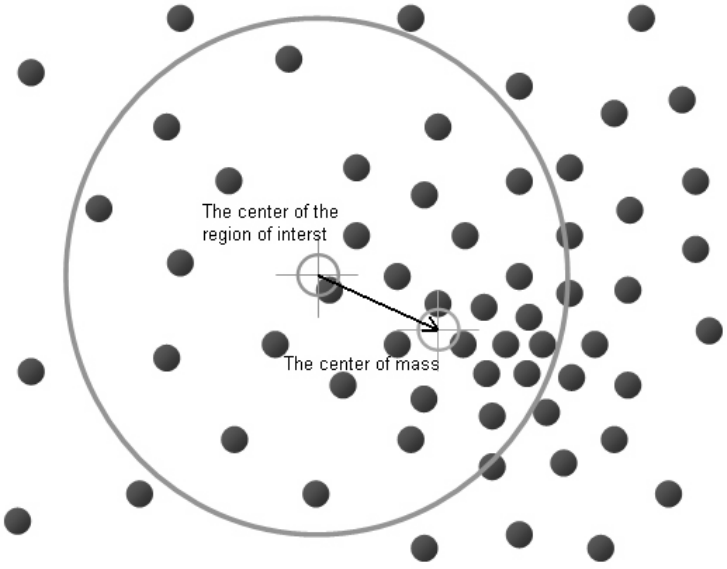
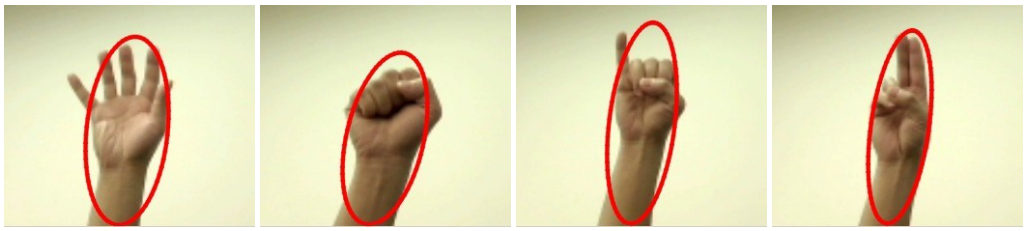


Figure 10 The mean shift algorithm [42]



(a) (b) (c) (d)

Figure 11 CAMShift based hand tracking [43]

Lowe in [44], proposed a method called Scale Invariant Feature Transform (SIFT) was introduced to extract distinctive invariant features from images that can be used to carry out reliable matching between various views of an object or a scene, as shown in Figure 12. The SIFT features utilize scale-space extrema in the difference-of-Gaussian (DoG) function, which convolves with the image as the key points that are invariant to image scaling. By assigning a consistent orientation for every key point that depend on local image features, the key point descriptor can be represented in relation to this rotation. It can thus attain invariance to image rotation, as is shown in Figure 12 for the rotated toy train in the image.

Figure 12 Training images, left side. Results of recognition, right side[44]

In addition to the invariance against image scaling and rotation, the SIFT features are also partially invariant against changes in lighting conditions and 3D camera viewpoints. In [44], Lowe also described a method for object recognition

using these features. The recognition is achieved by comparing individual features to a database of known objects using a fast nearest-neighbour approach. A Hough transform is then used to discover clusters belonging to a single object, and then verification is executed by using a least-squares solution for consistent pose parameters. This recognition method can robustly recognize objects with cluttered backgrounds and occlusion, and the method can achieve real-time performance for low resolution images.

The object detection algorithm proposed by Viola and Jones in [45] was designed to be real time applications and reliable for many objects in many difficult conditions. This algorithm was originally used for face detection. Viola et al. in their approach used the concept of an integral image. This concept is used to compute a rich set of image features. They also developed feature selection algorithm that is based on the AdaBoost (Adaptive Boost) learning algorithm. This algorithm is a different from regular boosting algorithm, because it can adaptively select the best features at each step and combine a series of weak classifiers into strong classifiers. Limited research [46] [47] [48] has been conducted to extend the method to hand detection and sign recognition.

In [49] [50] [51], researchers used the Principal Component Analysis (PCA) approach for hand detection. The primary linear method for dimensionality reduction is the principal component analysis. It carries out a linear mapping of the information to a lower dimensional space to maximize variance of the information in the low dimensional representation. The correlation matrix of the data is built, and the eigenvectors are calculated. The eigenvectors that are related to the largest eigenvalues (the principal components) will be utilized to reconstruct a large fraction of the variance of the original information. Furthermore, the first few largest eigenvectors can usually be interpreted in terms of the large-scale physical behavior of the system. The original space with dimension of the number of points has been decreased (with information loss, but ideally keeping the most significant variance) to the space spanned by a few eigenvectors.

In [52], Morguet et al. used a Hidden Markov Model (HMM) to find hand signs. This was based on an improved version with the context free keyword

spotting method. Their approach was to use color histograms to segment. They could then easily calculate the sequence of binary images, which contained only hand shapes. All signs were performed by an individual entity. In [53], Wang et al. also used the Hidden Markov Model to recognize dynamic hand signs. They used an AdaBoost algorithm and histograms of gradient orientation (HOG) to detect hands and to record the trajectory of a palm's center. Later, they did quantization of local and global features and extracted the discrete path vector feature from the sign path. Finally, they classified signs by using the Hidden Markov Model, which lacked performance under different illumination conditions.

In [54], Suk et al. conducted hand sign recognition in a continuous video stream by using the dynamic Bayesian network model. This model has followed different techniques like skin extraction, modeling and motion tracking. To define a cyclic sign network, it advanced sign models for more than one hand sign. It is developed only for continuous sign recognition. The recognition rate for the continuous stream of signs has increased up to 84%, with the precision of 80.77%. David et al. in [55] proposed a dynamic trajectory segmentation approach for dynamic hand sign recognition. The approach used a tensor voting technique to filter and construct a smooth trajectory. It deals with locally linear- trajectories that allow them to make decisions that are firmly based on corresponding modes in radon space. Later, they encoded the entire trajectory through the use of direction sequences. This process helped them define the number of possible signs in a human-computer interaction.

Ramamoorthy at al. in [56], proposed a combination of static shape recognition, by using contour discriminant analysis. Ramamoorthy et al. detected a hand by using a Kalman filter and a Hidden Markov Model to attain temporal characterization. They claim that this system works well in a cluttered background. It also uses skin color detection, for static shape recognition and tracking.

2.1.3 Machine Learning Algorithm Based Approaches

Learning algorithm based approaches stem from the artificial intelligence community. Their common trait is that recognition accuracy can be increased through training. The most common approaches of this method are neural networks, the Hidden Markov Model (HMM) and instance based learning.

Approaches based on this method comprise of two sets: one set is used as a training set and other is used as a testing set. Initially, the training set is used by an automatic classifier for learning the differentiating features of signs. The testing set is later used to test the accuracy of the automatic classifier. The machine learning based method concentrates on optimising either a group of parameter values, with regard to a group of features, or a group of induced rules with regard to a group of attribute-value pairs.

Machine learning based techniques have been successfully used in numerous applications, including in medical diagnosis, internet search engines, bioinformatics, credit card fraud detection, speech recognition, handwritten recognition, stock market analysis and classification of DeoxyriboNucleic Acid (DNA) sequences. The main concept of machine learning is to make the system capable of learning patterns from available information and then to later recognize these patterns.

The first successful machine learning method was the artificial neural networks (ANN), proposed by [57]. ANN method has two main advantages, which are the existence of a well-tested implementation system and a relatively fast classification speed, as compared to training time. Other machine learning approaches also have been used for hand tracking and hand recognition, such as Bayesian networks [58] [59] [60], AdaBoost [61], decision tree [62] [63] and fuzzy models [64] [65] [66]. Here we discussed some important methods.

Artificial Neural Networks (ANN)

An artificial neural network [70] is an information processing model that is motivated by the biological nervous system, such as the brain, to process information. While the neuron acts as the fundamental functional unit of the brain,

the neural network uses the node as its fundamental unit. The nodes are connected by links, and the links have an associated weight that can act as a storage mechanism. Each node is considered a single computational unit containing two parts. First part is the input function. Its job is to compute the weighted sum of its input values. The second part is the activation function. Its job is to transforms the weighted sum into a final output value.

Neural networks generally have two basic topologies:

- A feed-forward structure
- A recurrent structure

A feed-forward network can be considered a directed acyclic graph, while a recurrent network has an arbitrary topology. The recurrent network has the advantage over a feed-forward network in that it can model systems with state transitions. However, recurrent networks require more complex mathematical descriptions and can exhibit chaotic behavior. There is no restriction on the number of layers in the network in any topology.

These multilayered networks offer more representation power, but have to compromise on the cost of more complex training. There is no communication with the outside world of nodes that comes in between the input and output nodes. These in between nodes also cannot be directly observed from the input or output behavior of the network [67]. There are two types of training in neural networks.

- Supervised learning
- Unsupervised learning

Supervised learning trains the network by providing matching input and output patterns. This trains the network in advance and results in the network not learning while it is running**. Unsupervised learning** may also be referred to as self – organization, which trains the network to respond to clusters of patterns within the input. It does not have any advance training. This means that the system must develop its own representation of the input, since no matching output is provided [68]. There are scenarios where we can use combination of both learning strategies, as neither of these are mutually exclusive.

A. R. Varkonyi-Koczy et al. in [69] conducted modeling of hand postures and signs. Later, they developed a recognition system of hand signs to communicate with a smart environment. They used fuzzy neural networks for the recognition of hand signs. Their system was able to recognize a user's hand sign analyzing the sequence of detected hand postures. They did not recognize dynamic signs. Consequently, they lacked the ability for humanoid interaction in a smart environment.

Neural networks are a useful method for recognizing hand postures and signs. However, they have distinct disadvantages:

- Different configurations of a given network can give very different results, and it is difficult to determine which configuration is best without implementing them.
- The considerable time involved in training the network.
- The entire network must be retrained in order to incorporate a new posture or sign. If the posture/sign set is known beforehand this is not an issue, but if postures and gestures are likely to dynamically change as the system develops, a neural network is probably not appropriate.

Instance Based Learning

Instance based learning [71] [72] is another machine learning based approach. With instance based learning, the training data is used as a database with which to classify other instances. An instance is a vector of features of the entity to be classified. For example, during gesture recognition, a feature vector might be the position and orientation of the hand and the bend values for each of the fingers.

Instance based learning methods also have techniques that shows instances as pints in Euclidean space. The training phase of the algorithm involves storing a set of representative instances in a list of training examples. For each new record, the Euclidean distance is computed from each instance in training example list and the closest instances to the new instance are returned. The new instance is then classified and added to the training example list, so that training can be continuous.

Instance based learning methods are very simple. However, they do have some disadvantages, such as:

- They require large amount of primary memory as training set increases.
- Response time issues may arise due to a large amount of computation at instance classification time.

Support Vector Machine (SVM)

The support vector machine was developed by Vladimir N. Vapnik [73]. It was later enhanced by Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik [74]. It is a set of associated supervised learning techniques applied for classification and regression. SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification regression or other tasks. It optimally divides the data into two groups. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (functional margin), since in general, the larger the margin, the lower the generalization error of the classifier. Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space.

In SVM technique, there is a predictor variable known as attribute and there is a transformed attribute that is applied to identify the hyperplane, known as feature. The task of selecting the most appropriate representation is called as feature selection. A group of features that describe one case is known as vector. The purpose of SVM modeling is to find the optimal hyperplane that divides clusters of vectors in such a way that cases with one class of the target variable are on one side of the plane and cases with the other classes are on the other side of the plane. The vectors near the hyperplane are the support vectors. While SVM were initially proposed as binary classifiers, other methods that deal with a multi-

class problem as a single "all-together" optimization problem exist. However, they are computationally costlier than solving several binary problems [75].

There are number of approaches proposed for the decomposition of the multiclass problem into several binary problems using SVMs as binary classifiers. Kernel SVMs are available in many machine learning toolkits such as Library for support vector machine (LIBSVM) [76]. This library has support for multi-class classification and it uses a one-against-one (OAO) approach for multi-class classification in SVM [77].

For the M-class problems, whereas M being greater than 2, the one against one (OAO) approach creates M(M-1)/2 two-class classifiers, using all the binary pair-wise combinations of the M classes. Each classifier is trained by using the samples of the first class as positive examples and samples of the second class as negative examples. To combine these classifiers, the Max Wins method is used to find the resultant class by selecting the class voted by the majority of the classifiers [78].

2.1.4 Rule Based Approaches

In rule based approaches, one's sign features are compared to manually encoded rules. If any feature or features during comparison justify the rule, the resulted sign will be provided as output. Cutler et al. in [79] proposed a rule based technique to recognize an action depending on a set of conditions in their view based approach to sign recognition.

Cutler and Mu-Chun respectively in [80] [81] predicates related to low-level features of the motion of the hands are defined for each of the actions under consideration. When a predicate of a sign is satisfied over a fixed number of consecutive frames the sign is returned.

Hassan et al. in [82] proposed a rule based technique for analyze and understand semantically the motion of the trajectories of the human activity. They categorized the detected trajectories according to the distances between their adjacent points. Their proposed system can be used for any point-based tracking system.

Chang et al. in [83] proposed a fuzzy rule base approach in human activity recognition. They used fuzzy rule base for posture sequence processing and recognize action with best matches the posture sequence in the fuzzy rules. Fuzzy rule approach can not only combine temporal sequence information for recognition but also be tolerant to variation of action done by different people.

Zou et al. in [84] used Deterministic Finite State Machine (DFSM) to detect hand motion and then apply rule based techniques for gesture recognition. He defines gestures into two category based on motion linear and arc shaped gestures. A major problem with rule-based approaches is that they rely on the ability of a human to encode rules. In many cases the appropriate rules may not be intuitive especially when dealing with high-dimensional feature sets.

2.1.5 Syntactic Approaches

Computer vision problems that have complex patterns and activities are very hard to be address by statistics approaches. Sometimes, numeric measurements might not be enough to represent the complex structures of these patterns and activities. These situations are well handled by syntactic approaches [85]. The elementary parts used to syntactically describe a complex pattern or an activity are called primitives. There are few principles that need to follow to identify the primitives [86]. These include that:

- The number of primitive types should be small
- The selected primitives must be able to form an appropriate representation of object
- It should be easy to segment the primitive from the image
- Primitives should correspond with significant natural elements of the object structure being described
- It should be easy to recognize primitives using some statistical pattern recognition method.

After the extraction of primitives, a grammar representing a set of rules must be defined so that different patterns and activities can be constructed based on the

extracted primitives. A mathematical model for the grammar's structure can be defined as:

$$G = [Vt, V_n, P, S]$$

Whereas:

V_t is a set of terminals, which are the most primitive symbols in the grammar.

Vn is a set of non-terminals, which are symbols composed by a collection of terminal s and / or non-terminals. There is a condition that Vt and Vn are disjoint alphabets.

P is a finite set of production rules, which is a transformation from one sequence of terminals and/or non-terminals to another sequence of terminals and non-terminals.

S is a start symbol, which shows where all valid sequences of symbols we want to be able to produce can be derived from.

Figure 13 shows the Chomsky hierarchy of grammars. According to Chomsky, grammars can be divided into four types ordered from general such as unrestricted grammar to specific such as regular grammar [87].

- Unrestricted Grammar: Type 0
- Context-Sensitive Grammar: Type 1
- Context-Free Grammar: Type 2
- Regular Grammar: Type 3

Unrestricted Grammar: Type 0

Unrestricted grammar includes all types of grammar. It has no limitation for the substitution rules. It can have any strings on either the left side or the right side of the substitution arrow.

Context -Sensitive Grammar: Type 1

The substitution rule of context-sensitive grammars is:

$\alpha A \beta \rightarrow \alpha \gamma \beta$

Where A is a nonterminal, α, β and γ are strings of terminals and nonterminals. α and β can be empty strings but γ must be nonempty. This substitution rule can be understood as "A can be replaced by γ in the context of α, β".

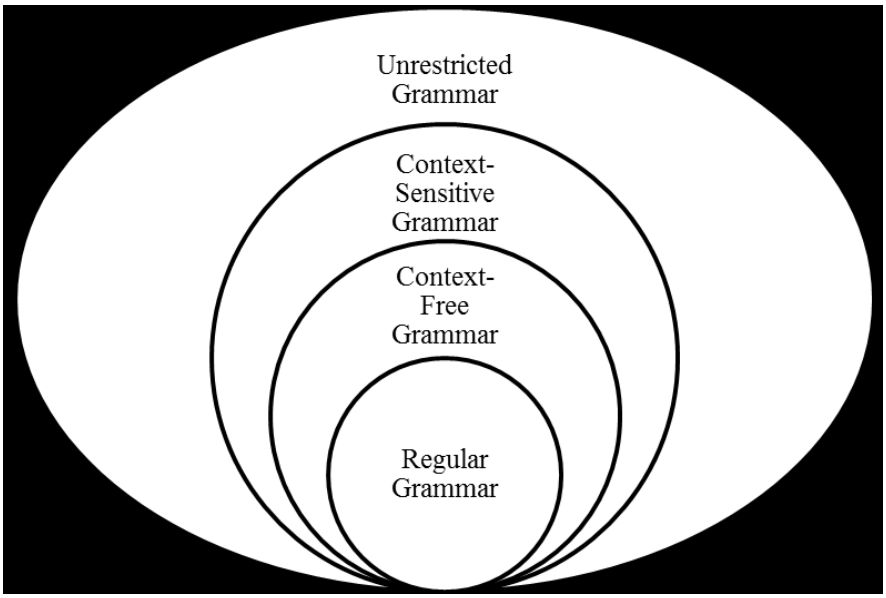


Figure 13 The Chomsky hierarchy of grammars.

Context-Free Grammar (CFG): Type 2

The substitution rule of context-free grammars (CFG) is:

$A \rightarrow \gamma$

Where A is a nonterminal, γ is a nonempty string of terminals and nonterminals. This substitution rule limits the left side consisting of only a single nonterminal. The context-free grammar allows the nonterminal A to be replaced by the string γ independently of the context where A appears. One example of context free grammars is: $G = (V_N, V_T, P, S)$, where $V_N = \{S, A, B\}$, $V_T = \{a, b\}$, and P:

(1)$S \rightarrow aB$ (2)$S \rightarrow aB$

(3)$A \rightarrow aS$ (4)$A \rightarrow bAA$

(5)$A \rightarrow a$ (6)$B \rightarrow bS$

(7)B → aBB (8)B → b

This grammar defines a language L(G), which is the set of strings consisting of an equal number of a's and b's such as ab, ba, abba and bbaa.

Regular Grammar: Type 3

The substitution rule of regular grammars is:

A→ αB or A → β

Where A and B are nonterminals, α and β are terminals or empty strings. Besides limiting the left side consisting of only a single nonterminal, this substitution rule also restricts the right side: it may be an empty string, or a single terminal symbol, or a single terminal symbol followed by a nonterminal symbol, but nothing else. The languages generated by regular grammars are called regular or finite state languages. One example is: G = (V_N, V_T, P, S), where V_N = {S,A}, V_T = {a, b}, and P:

S → aA

A → aA

A → b

This grammar defines a language L(G) = {a^nb|n = 1, 2, . . .}, which includes strings such as ab, aab, aaaab.

Figure 13 shows that the Chomsky hierarchy is inclusive. That means that every regular grammar is context-free, every context-free grammar belongs to unrestricted grammar. Even though unrestricted grammars are the most inclusive grammar, they are too general to be useful. It cannot be decided whether a specific string is generated by an unrestricted grammar. Unrestricted grammar is more powerful than context-free grammar and regular grammar. However, these two restricted grammars are most often used, because parser for them can be efficiently implemented [88]. The finite state machine is used to recognize all regular languages. For context-free grammar, there are many parsers to recognize the corresponding languages.

Hand et al. [89], proposed a syntactic approach to understand hand signs. They defined a set of signs as terminals of the language. These include some hand

postures like HO for hand open, HF for hand fist and IF for index finger outstretched. They also added multiple hand movement terminals, such as MU, MD for move up and down, ML, MR for move left and right and MT, MA for move towards and away. After the decision of terminals, a set of production rules are defined. Those are:

<Gesture> := = <Pick>|<Point>|<Translate>|<Undo>|<Redo>|<Stop>

<Pick> := = <ChooseLocation><Grab><Drag><Drop>

<ChooseLocation> := = TIF-Motion <ChooseLocation>

<Grab> := = TIFC

<Drag> := = TIFC-Motion <Drag>

<Drop> := = TIF

<Point> := = <MovePoint> <SelectPoint>

<MovePoint> := = IF-Motion <MovePoint>

<SelectPoint> := = TIF

<Translate> := = FTFO <Direction>

<Direction> := = ML|MR|MU|MD|MT|MA

<Undo> := = IMFO MA

<Redo> := = IMFO MT

<Stop> := = HF HO

The above mentioned production rules show that if the user wants to perform a "stop" command, he must make a fist and then release it. In order to drag any object, he must hold the thumb and index finger closed, and continue to move it.

Derpanis et al. [90], proposed an approach to represent complex gestures in terms of their primitive components. They decomposed dynamic gestures into static and dynamic components, in terms of three sets of primitives. Those components are hand shape, location and movement. Their proposed algorithm can recognize gesture movement primitives given data captured with a single video camera. By working with a finite set of primitives, which can be combined in a wide variety of ways, their approach has the potential to deal with a large vocabulary of gestures.

Ryoo and Aggarwal [91], proposed an approach for the automated recognition of complex human activities. They use a context-free grammar based representation scheme to represent composite actions and interactions. The context-free grammar describes complex human activities based on simple actions or movements. Human activities are classified into three categories:

- Atomic action
- Composite action
- Interaction

The system was tested to represent and recognize eight types of interactions: approach, depart, point, shake-hands, hug, punch, kick, and push. The experiments resulted in a high recognition rate to recognize sequences of represented composite actions and interactions.

2.1.6 Local Feature Approaches

Recently, significant research has been conducted in local-spatio temporal features and bag of features (BOF). This approach achieved significant performance. This approach represents gestures with a collection of independent local spatio-temporal regions. The success of such methods can be attributed to their relatively independent representation of events, which has better tolerance to certain conditions in the video, such as illumination, occlusion, deformation and multiple motion.

Laptev et al. in [92], propose the concept of spatial interest points into the spatio-temporal domain and show that the resulting features often reflect interesting events that can be used for a compact representation of video data, as well as for its interpretation. They described an interest point detector that finds local image values in space-time characterized by high variation of the image values in space and non-constant motion in time. They were first to propose a space-time interest point by extending a 2D Harris-Laplace detector. They compute a spatiotemporal second-moment matrix at each spatio-temporal point with different spatial and temporal scales, a separable Gaussian smoothing function and space-time gradients. The authors also apply an optional mechanism

to select different spatio-temporal scales. The authors showed that video representation by spatio-temporal interest points enables the detection and pose estimation of walking people in the presence of occlusions in a highly cluttered and dynamic background.

Dollar et al. in [93], showed that the direct 3D counterparts to commonly used 2D interest point detectors are inadequate; they describe an alternative. Anchoring off of these interest points, they devised a recognition algorithm based on spatio-temporally windowed data. To produce denser space-time feature points, researchers used a pair of 1D Gabor-filter to convolve with a spatial Gaussian to select local maximal cuboids. Interest points are the local maxima of the response convolution. They proved that the use of cuboid prototype gave rise to an efficient and robust behaviour descriptor.

Recently, Wang et al. are focusing on dense sample points [94] and trajectories [95][96] in order to enhance the performance. Most feature detectors are extended from computationally costly image feature extraction methods. Considering the huge amount of data to be processed, this is very limiting factor. As compared to sparse interest point-representation, dense sampling methods capture most information by sampling every pixel in each scale.

Scovanner et al. [97], proposed a technique of random sampling on video at different times, locations and scales to extract feature points. They created an extension of a 2D SIFT descriptor into a 3D SIFT descriptor to represent spatio temporal patches. Wang et al. in [94], conducted a comparison of some formerly proposed space time features. Their evaluation results show that dense sampling at regular space-time grids outer performs state-of-the-art interest point detectors.

Messing et al. in [96], proposed a method to get feature trajectories using Kanade-Luca-Tomasi (KLT) Tracker. Researchers in this method applied uniform quantization in log polar coordinates to represent feature trajectories with varying length. They used eight bins for direction and five bins for magnitude. Another researcher in [98][99], proposed a method to use fixed length feature trajectories instead of varying length feature trajectories for action classification. They clustered the trajectories from the video by K-means. For each cluster center, an

affine transformation matrix is computed. The final trajectory descriptor contains information about both displacement vectors and elements of the affine transformation matrix for its assigned cluster center.

Feature Descriptor

Feature descriptors are vital for the performance of the video recognition. A feature descriptor is computed in a local neighbourhood to capture shape and motion information for each spatio-temporal feature point. Laptev et al. in [100], examined a set of local space-time descriptors for representing and recognizing motion patterns in video. Particularly, they computed and evaluated single and multi-scale higher-order derivatives (local jets), histograms of optical flow (HOF) and histograms of spatio-temporal gradients. Their computation showed that in terms of local position, dependent histograms of either spatio-temporal gradients or optical flow give significantly better results, as compared to global histograms, N-jets or principal component analysis of either histogram of gradient or HOF. They calculated histograms of gradient and HOF for each cell over an M x M x M grid layout. They used PCA to reduce the dimension of features, which are computed by a concatenated optical flow or gradient components of each pixel.

Laptev et al. in [101], proposed an approach for automatically collecting training data for human actions and has shown that this data can be used to train a classifier for action recognition. They combined histograms of oriented spatial gradient (HOG) and HOF to include both local motion and appearance information. Dalal and Triggs in [102], introduced HOG descriptor for human detection. Their method was motivated by the SIFT descriptor. They computed the gradients and dense optical flow and divided each local region into a grid of N x N x M cells. They computed four bin HOG histograms and a five bin HOF histogram for each cell. Later, they concatenated them into the final HOG / HOF descriptor. Their experiments showed that HOG descriptors notably did better than existing feature sets for human detection.

Klaser et al. in [103], proposed a local descriptor, known as histograms of oriented 3D spatio-temporal gradients (HOG3D) descriptor. Researchers built it

on 3D oriented gradients. They also proposed a generic 3D orientation quantization, which is based on regular polyhedrons. The 3D histogram of oriented gradients for the 3D patch is formed by concatenating gradient histogram of all cells. While it is efficient to compute 3D gradients, the orientation quantization with polyhedron for each sub-blocks is relatively expensive to compute.

Bag-Of-Features (BOF)

Bag-of-features idea is initially derived from "bag-of-words (BOW)" representation for text classification and document analysis [104] [105] [106] [107] [108] [109]. By extracting representative words in the training set of various sentences, a dictionary is formed. Meaningful sentences are substituted by the frequency of the occurrence of the words in the dictionary which is regarded as a 'bag'. When a new sentence comes, it can be coded by the dictionary and classified into a specific category by computing its similarity with trained 'bags'.

Recently, BOF gained considerable popularity in object classification from images and action recognition from video data, because of its simplicity, robustness and good performance. BOF is effectively applied to object and natural scene categorization. It models videos as collections of local spatio-temporal patches. The BOF approach has a spatio-temporal patch, which is represented with one of the feature descriptor as a feature vector. A good feature descriptor or feature vector should have the ability to handle intensity, rotation, scale and affine variations, to some extent. The final step for the BOF approach is to convert vector represented patches into a "codebook". A vocabulary of prototype features are called "visual words" or a "codebook". This is accomplished by applying a clustering algorithm (e.g. K-means) on feature vectors computed from training data. A video is represented as a histogram of occurrences of local features by quantizing the feature vectors to the closest visual word.

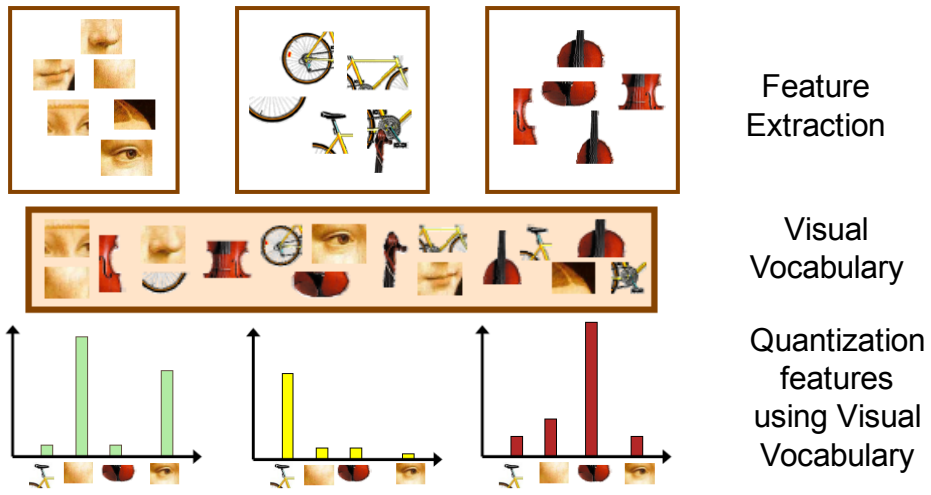


Figure 14 Bag-of-Feature approach steps

Figure 14 shows the different steps of BOF approach. The BOF approach lacks to maintain order of features. It only contains statistics of unordered features from the image sequences and any information of temporal relations or spatial structures is ignored. This can cause a problem when recognizing dynamic motions or dynamic gestures, because the approach is not able to maintain event order, such as an order of gesture start and gesture end. Researchers have applied many different methods to maintain orders. Hamid et al. in [110], proposed an unsupervised method for anomalous detection as overlapping n-grams of actions. n-grams can preserve the temporal order information of events during overlapping. It causes the dimensionality of the space to grow exponentially as n increases.

Laptev et al in [101] extend image representation of spatial pyramid to the spatio-temporal domain. They divide a video into a grid of coarse spatio-temporal cells. The entire video is then represented by the ordered concatenation of the per-cell BOF models. Such ordered concatenation adds to global structural information. A "greedy" approach is used to learn the best combination of overlaid grids and feature types per action class.

2.2. Calibration

Algorithm calibration is a comparison of measuring algorithm against a standard instrument of higher accuracy to detect, correlate, adjust, rectify and document the accuracy of the algorithm being compared. Although the exact procedure may vary from algorithm to algorithm, the calibration process generally involves using the instrument to test samples of one or more known values called "calibrators." The results are used to establish a relationship between the measurement technique used by the instrument and the known values.

The calibration of an algorithm is checked at several points throughout the calibration range of the algorithm. The calibration range is defined as "the region between the limits within which a quantity is measured, received or transmitted, expressed by stating the lower and upper range values." The limits are defined by the zero and span values. The zero value is the lower end of the range. Span is defined as the algebraic difference between the upper and lower range values. The calibration range may differ from the instrument / algorithm range, which refers to the capability of the instrument / algorithm. Calibration is required to check the accuracy of algorithm.

Calibration can be used for different tasks like robot calibration [113] [114] [115] [116] [117], camera calibration [111] [112], algorithm calibration etc. Gesture estimation has close relationship with human body move estimation or the pose estimation of articulated objects in general. Human body movement estimation is a more intensive research field. Many algorithms used in gesture tracking have a lot of similarities to algorithms proposed previously for human body move estimation. However, there are also many differences in operation environments and related applications. Our research in this thesis, addressing the problem of dynamic gesture recognition and its calibration. Humans have tendency to repeat gestures every time with variations in time and space. We consider this restriction of humans in our chapter 4 and did calibration of our algorithm. We first created videos with human hand as subject, then test them against all as mentioned parameters and scenarios in chapter 4. Then, we created same number of videos under same conditions with robot hand to verify and calibrate our algorithm as

robot hand has ability of consistency of repeating gestures number of time by using same space and time.

2.3. Gesture Recognition Technologies

Hand sign and more recently arm gesture recognition is one of the major areas of research for the engineers, scientists and bioinformatics. Hand gesture recognition is the natural way of human-robot and inter-robot interaction and today many researchers in the academia and industry are working on different technologies to make interactions more easy, natural and convenient without wearing any extra device. However, human-robot interaction using gestures provides a formidable challenge. For the vision part, the complex and cluttered backgrounds, dynamic lighting conditions and a deformable human hand shape, wrong object extraction can cause a robot/computer to misunderstand the gesture. If the robot/computer is mobile, the gesture recognition system also needs to satisfy other constraints, such as the size of the gesture in the image and adaptation to motion. In addition, to be natural, the robot/computer must be person independent and give feedback in real time.

For hand posture and gesture recognition system different technologies are used for acquiring input data. Various known companies are doing research to produce technologies based on human-computer/robot interaction and gesture recognition. Ford and Intel announced they have been collaborating on a new research initiative that explores new functions for the connected car, such as allowing drivers to remotely view the interior of their car using a smartphone, or a facial recognition system that would allow the owner to gain access and drive their vehicle, according to ZDNet. Named Mobile Interior Imaging, or Project Mobii, the joint research project looks at how interior-facing cameras could be coupled with sensor technology and data.

Juniper Research [118], has for many years tracked leading edge technological developments within the field of mobile and consumer electronic devices. Biometrics and Human Interface Technologies are amongst the latest innovations which they have explored in-depth. This new research explores the

opportunities for the various technologies underpinning gesture-based, touchless and biometric functionalities in the mobile device. Juniper Research argues that while human interface technology will not abandon touch commands entirely, touchless and biometric interfaces will continue to play an increasing role in enhancing user experience and handset security. This analysis singles out touchless screen scrolling, such as Samsung's Smart Scroll feature and the integration of Touch ID into Apple Pay as examples of growth areas in which these technologies will likely experience more deployments.

Gesture recognition has been described by Biometrics Research Group, Inc. [119] as "the mathematical interpretation of a human motion by a computing device. Gestures can originate from any bodily motion or state but commonly originate from the face or hand", while touchless sensing technologies enable users to interact with a device without touching it. Touchless sensing can be divided into two broad markets based on the applications, namely touchless sanitary equipment and the touchless biometric market. The touchless sanitary equipment market is segmented by major product categories such as: faucets, soap dispensers, trash cans, hand dryers, paper towel dispenser, and flushes. The touchless biometric market is segmented by different modalities, including: face, iris, voice, and touchless fingerprint biometrics.

Biometrics Research Group, Inc. [119] report that technologies that track gesture movements will play a large role in future mobile applications and devices. According to a recent research, the newest smartphone from Samsung have a gesture tracking feature that will allow its users to scroll right and left a page without having to touch the screen.

Microsoft developed a device name Kinect for gaming purpose. This device has ability for human-computer interaction. Kinect is discussed briefly in the following section.

2.3.1 Microsoft's Kinect

Kinect is an add-on device, shown in Figure 15, for the Microsoft gaming system that enables users to control games, movies and music with physical motion or

voice commands and without the need for a separate input controller like a joystick or keyboard. The controller-free gaming environment provided by Kinect makes it possible for sensors to process basic gestures, facial characteristics, sounds and even full body motion activities such as jumping and kicking [120] [121] [122].

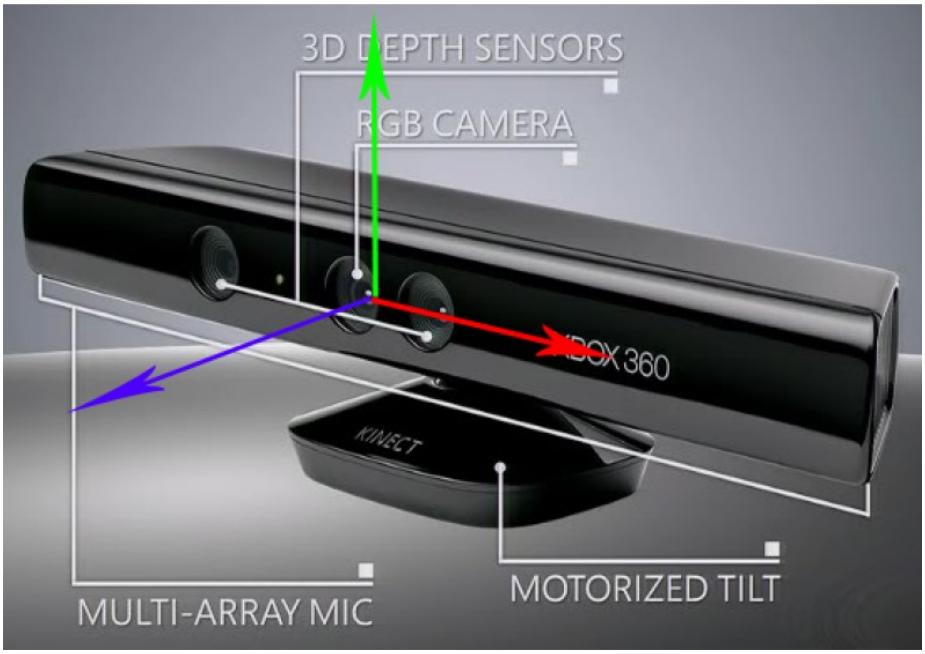


Figure 15 The Kinect device. The z–axis is pointing out of the camera [123]

It employs cameras and microphones to detect and track the motions of a user standing in front of it and permits voice control of games and video content. The Kinect is capable of concurrently tracking up to six players, involving two active users for movement analysis with a feature extraction of 20 joints per user.

Gesture recognition is a fundamental element when developing Kinect-based applications (or any other Natural User Interfaces). Gestures are used for navigation, interaction or data input. The most common gesture examples include waving, sweeping, zooming, joining hands, and much more. Unfortunately, the current Kinect for Windows SDK does not include a gesture-detection mechanism out of the box [124]. Kinect has limitation that user has to repeat same gesture at least three to four times to understand successfully that gesture. It limits it usage

for Dynamic Gesture Language Recognition where one iteration is one word and more than one iterations are that respective number of words. It is still a challenge to employ Kinect for hand gesture recognition because of the low-resolution of the Kinect depth map of only 640×480 pixel.

Kinect has an infrared camera and a PrimeSense sensor to calculate the depth of the object, and an RGB camera to capture frames. The depth images and RGB image of the object could be got simultaneously. This 3D scanner system named Light Coding uses a variant of image-based 3D reconstruction [125]

While Kinect can track a large object such as the human body, it has problems when detecting and recognizing smaller objects such as a complex articulated human hand which occupies a very small region of the image. As a result, the segmentation of the hand is inherently imprecise, and this may considerably affect its recognition [126]. Beside above mentioned limitations, Kinect software permits advanced human body tracking, facial recognition and voice recognition [127].

2.4. Gesture Applications

Gesture recognition systems have been used for large variety applications on different domains as a human friendly type of interaction. Here we discuss some new applications of vision based gesture recognition in recent years. This will give an indication of its prospects in future.

2.4.1 Touch-Based Devices Alternative

Smart phones, touch screen computers, shopping mall touch displays and tablets are types of devices that can take over our everyday life by replacing traditional input devices (mouse, keyboard) by touch to control devices. Currently, these vision based gesture recognition techniques make it possible to give users a touch-free solution.

2.4.2 Sign Language Recognition

Sign language is one of the most promising sub-fields in gesture recognition research. Effective sign language recognition would grant the deaf and hard-of-hearing expanded tools for communicating with both other people and machines. In order to realize this technology, researchers must devise methods to capture and record both individual hand positions and the motions that create them for a constant, fluid, and accurate interpretation of sign language. Since sign language is used for interpreting and explaining a certain subject during a conversation, significant research has been conducted on this subject.

The American Sign Language (ASL) in [128], is recognized as using boundary histogram, MLP neural network and dynamic programming matching. Kouichi and Hotomi in [129], recognized Japanese sign language (JSL) using Recurrent Neural Network. Over time, all manner of machines may come to be operated through natural gestures, making human interaction with their tools and world more dynamic and fluid than ever before in modern times. The implementation of a human-like gesture language is still facing many problems, due to the complexity of gesture recognition, and naturally, there is a high variation in the performance of user gestures, such as their physical features and environmental conditions.

2.4.3 Human-Robot Communication

Human communication with robots through using hand signs and arm gestures has been adopted by many popular robotic applications [130] [131]. Ghobadi et al. in [131], proposed a system that control the robot using hand pose signs. They map the first five mathematical numbers with different commands to perform.

Dr Robot Inc has introduced one of the first-generation commercially successful humanoid robot called “Hawk” [132] shown in Figure 16.

Figure 16 *Hawk* humanoid robot [132]

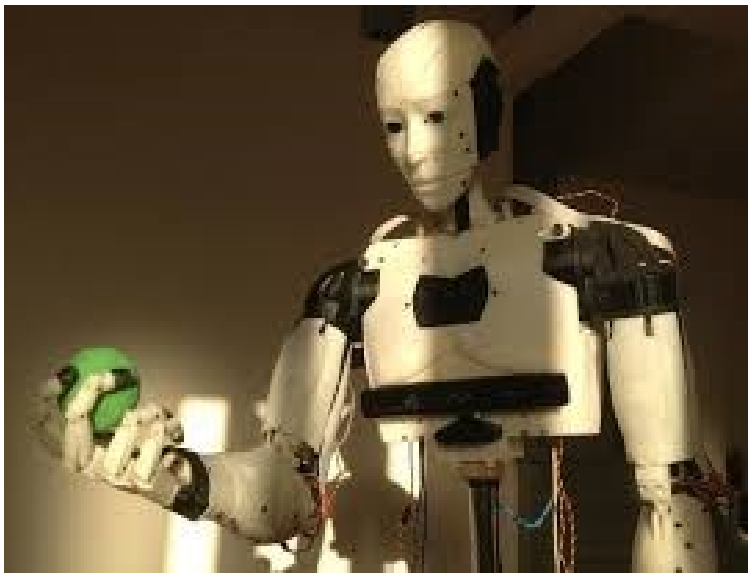


Figure 17 *InMoo*v open-source humanoid robot [133]

Gael Langevin, a French model maker and sculptor, has recently developed the InMoov robot [133], an *open source 3D printed life-size humanoid robot* shown in Figure 17

Replicable on any 3D printer with a 12x12x12cm area, it is conceived as a development platform for universities, laboratories, hobbyist, but first of all for Makers. Its concept, based on sharing and community, gives him the honor to be reproduced for countless projects throughout the world.

Petriu [1], discussed intelligent sensor environments and bio-inspired solutions for intelligent android perception and control for a new generation of humanoid robots (androids) which in order to naturally blend within human society should not only look as humans, but should also behave as much as possible as humans. Human operators and robotic appliances work together as symbionts, each contributing the best of their specific abilities. They proposed the use of common language and an underlying system of shared knowledge and common values for both human-machine and machine-machine communication. Petriu and Whalen in [2], analyzed that human computer-operator is beginning to act as a peripheral device for computers. They are most intelligent and dexterous peripheral device, to be sure, but a peripheral device, nonetheless. . It will be a symbiotic human-computer partnership in which each partner will lead in some cases and provide assistance in other cases

Chen et al. [3] proposed facial expression and gestures modalities for healthcare and smart environment applications in domain of human-computer interaction. The facial expression's recognition can help physically impaired people to control devices. Facial models can be used to the rehabilitation of people with impairments of facial muscles, and represent a useful tool in psychology in facial recognition and nonverbal communication areas. Application based on their proposed technique allows helping elderly and disabled individuals to live securely in their homes by monitoring their behavior.

Dr. Ivar Mendez says robotic and portable devices represent start of 'revolution' in health system [4]. Robots are delivering health-care by different ways in Saskatchewan. They are helping patients in several hospitals and clinics

in that province. In total, there are eleven medical robots and portable devices serving in Saskatchewan hospitals. Physicians can use smartphones to remotely control a robot and interact via video-link with a patient. Doctor successfully connect diagnostic equipment like stethoscopes, ultrasounds and electrocardiograms to see, touch and hear the patient. They use robots to serve patients and examine them remotely. When a person is seriously injured in any accident, the doctor can tele-stream into the ambulance, see the patient, advise the paramedic and conduct an ultrasound before the ambulance arrives at the hospital. Doctors can perform remotely their morning rounds or supervise a surgery by using robot stationed at hospitals [4].

Unhelkar and Shah [5] aimed to develop a mobile robot which can work alongside humans in automotive assembly lines. They use the Rob@Work-3 as the base platform for development of robotic assistant. This model robot able them to implement their algorithms by using Robot Operating System (ROS). They implemented basic sensing, actuation and safety capabilities in that robot. They faced reliability, usability, and maintainability issues of the robot. Lasota et al. [6] proposed motion planning technique that they call human-aware motion planning. In this technique, they used the prediction of human actions and workspace occupancy. This prediction helped to actively avoid potential motion conflicts that could arise in close-proximity human-robot interaction.

2.4.4 Multimodal Interaction and Virtual Environments

Virtual environments are one of the most popular applications of gesture recognition. Berry [134], proposed a system to control a virtual environment battlefield through the use of gestures. They not only used gestures to navigate, but also to select and move virtual objects in a battlefield.

Gesture recognition complements other communication modalities improving recognition accuracy and making up for one another's recognition mistakes. It is still an active area of research. Lucente et al. [135], have designed a multimodal interface that incorporates a vision-based hand posture and gesture

recognition solution with speech input for a number of different applications including vacation planning and real-estate purchases.

2.4.5 Medical Systems and Assistive Technologies

Gesture recognition has very important role in medical systems and its assistive technologies. It provides the user sterility needed to help avoid the spread of infection. Gestures can be used by doctors and staff to interact with medical instruments. It also helps patients with disabilities as part of their rehabilitation therapy, user adaptability and feedback. In this context, wheelchairs [136] have been enhanced through robotic and intelligent vehicles able to recognize gesture commands.

Figure 18 Surgeon using gestix to browse medical images [138]

In [137], Gratez et al. discussed different ways to integrate gestures with physician-computer interfaces. He illustrated a computer-vision system that allows doctors to carry out standard mouse functions such as pointer movement and button

presses with hand gestures that meet with the "intuitiveness" requirement. In [138], Wachs et al. designed a gesture-tracking machine named Gestix. This machine permits doctors to explore Magnetic Resonance Imaging (MRI) images in an operating room via a natural interface to meet with both "come as you are" and "intuitiveness". Its functionality is shown in Figure 18.

2.5. Conclusions

In this chapter, we provided a review of hand sign and arm gesture recognition approaches. The previous research is reviewed from two different dichotomies: the glove based hand posture recognition and vision based hand posture and arm gesture recognition. Our thesis focus is on vision based arm gesture recognition. Appearance based approaches, which have good real time performance, lack the ability to cover different classes of gestures, due to the simpler image features employed. Machine learning based approaches allow to increase the recognition accuracy by training.

Syntactic approaches describe complex patterns and activities with simple sub patterns and elementary parts. Local feature approaches are a significant part of the current era's topic of research. Local-spatio temporal features and the bag of features approach allow for a relatively independent representation of events, which has a better tolerance to certain conditions in the video, such as illumination, occlusion, deformation and multiple motion.

Chapter 3. Dynamic Arm Gesture Recognition for Human Subjects

Recently, a lot of research on pattern recognition is done using local spatio-temporal (ST) features and bag of features (BOF) which have already demonstrated significant increase in performance. In this approach gestures are represented as a collection of independent local spatio-temporal regions. The success of these methods can be attributed to their relatively independent representation of events which allows for better tolerance to certain conditions in the video scene, such as illumination, occlusion, deformation and multiple motion.

Bag-of-features idea was initially derived from the "bag-of-words (BOW)" representation used for text classification and document analysis [104] [105] [106] [107] [108]. By extracting representative words in the training set of various sentences, a dictionary is formed and a meaningful sentence pattern is defined by the frequency of occurrence of the words in the dictionary, which is regarded as a 'bag'. When a new sentence comes, it can be coded by the dictionary and classified into a specific category by computing its similarity to trained 'bags'.

Recently BOF gained great popularity in object classification from images and action recognition from video data due to its simplicity, robustness and good performance. BOF is effectively applied to object and natural scene categorization. It models video as collections of local spatio-temporal patches. In a standard BOF approach, a spatio-temporal patch is represented with one of the feature descriptors as a feature vector. A good feature descriptor or feature vector should have the ability to handle intensity, rotation, scale and affine variations to some extent. The final step for the BOF approach is to convert vector represented patches to "codebook". A vocabulary of prototype features are called "visual vocabulary" or "codebook". It is obtained by applying a clustering algorithm (e.g. K-means++) on feature vectors computed from training data. A video is represented as a histogram

of occurrences of local features by quantizing the feature vectors to the closest visual word.

As previously discussed BOF became very popular for object classification and because of its discriminative power. We are using BOF for gesture recognition on video data. Shi et al. [8], proposed a new approach of a 3-D multiscale parts model, which preserved the orders of events. Their model has a coarse primitive level 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 sub patches, they can successfully maintain an order of events. We used this novel approach that is based on hand and forearm movements. This idea enabled us to extend hand gestures to include forearm gestures. This evolution allows for a larger degree of freedom to have a variety of meaningful dynamic gestures based on arm in our dynamic gesture language vocabulary.

Our proposed dynamic gesture language recognition (DGLR) system has a few advantages over the previous systems. First, it is composed of two subsystems, namely the image processing (IP) module and the linguistic recognition system (LRS) module. LRS module will be discussed in chapter 5; 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. The BOF method has the ability to represent videos with statistical information of local features, with no requirements for the detection of humans, body parts or joint locations which is very challenging in uncontrolled realistic videos. DGLR is not only good for static postures, as the previous systems were, but also our choice of the local part model enables DGLR to recognize bare arm dynamic 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 gesture 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.

The current chapter gives a comprehensive account of the IP module which is first module of DGLR system.

3.1. The Dynamic Arm Gesture Recognition System Architecture Overview

To recognize the dynamic arm 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 dynamic arm gestures using a nonlinear SVM.

The arm consist of several pieces that together make it one of the most useful tools of the human body. These parts are *upper arm*: extending from the shoulder to the elbow, *elbow:* this is hinged joint allows the arm to swing 180 degrees at full extension, *forearm:* the forearm is the area between the wrist and the elbow, *wrist:* located in the upper hand, *hand:* palm with five fingers. Our selected dynamic gestures are mainly comprise of arm which is combination of upper arm, forearm, wrist, elbow and hand [139].

3.1.1 Overview of the Training Stage

First, we generate a database of dynamic gestures for the training stage. We didn't find any standard database of dynamic arm gestures on which we can apply our algorithm and compare our algorithm performance with others. So we did a thorough survey on previous researchers work and try to accumulate all possible considered parameters and scenarios in their work all together at one place in our work to make our experimental setup conditions tough and challenging. We consider a set of *twelve dynamic gestures*, namely the circular, goodbye, rectangular, rotate, triangular, wait, come, go, up, down, right and left 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 cycle. This gesture is a combination of hand, wrist, elbow, forearm and upper 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. This gesture as well involves the hand, wrist, elbow and forearm.

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 the hand, wrist, forearm, elbow and the upper 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 involves the hand, wrist and forearm.

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, wrist, forearm, elbow and an upper arm.

The *wait gesture* comprises using the wrist to move a hand in a forward and backward motion while the forearm remains static. This gesture is based on hand and wrist.

The *come gesture* starts by keeping the forearm straight down and hand facing front. Move forearm upward by bending elbow until hand and forearm reach complete straight up. This gesture is combination of hand, wrist, elbow and forearm motions.

The *go gesture* starts by keeping the forearm straight up and hand facing front. The forearm moves then downward by bending elbow until hand and forearm reach complete straight down. This gesture also involves combination of hand, wrist, elbow and forearm motions.

To complete the *up gesture*, one begins with one's hand positioned from middle-center and moves straight upward in same axis. This gesture involve hand, wrist, elbow, forearm and upper arm.

The *down gesture* starts from positioning hand and forearm at top-center. Later move hand straight downward direction until it reach at middle-center in same axis. This gesture also comprise hand, wrist, forearm, elbow and upper arm.

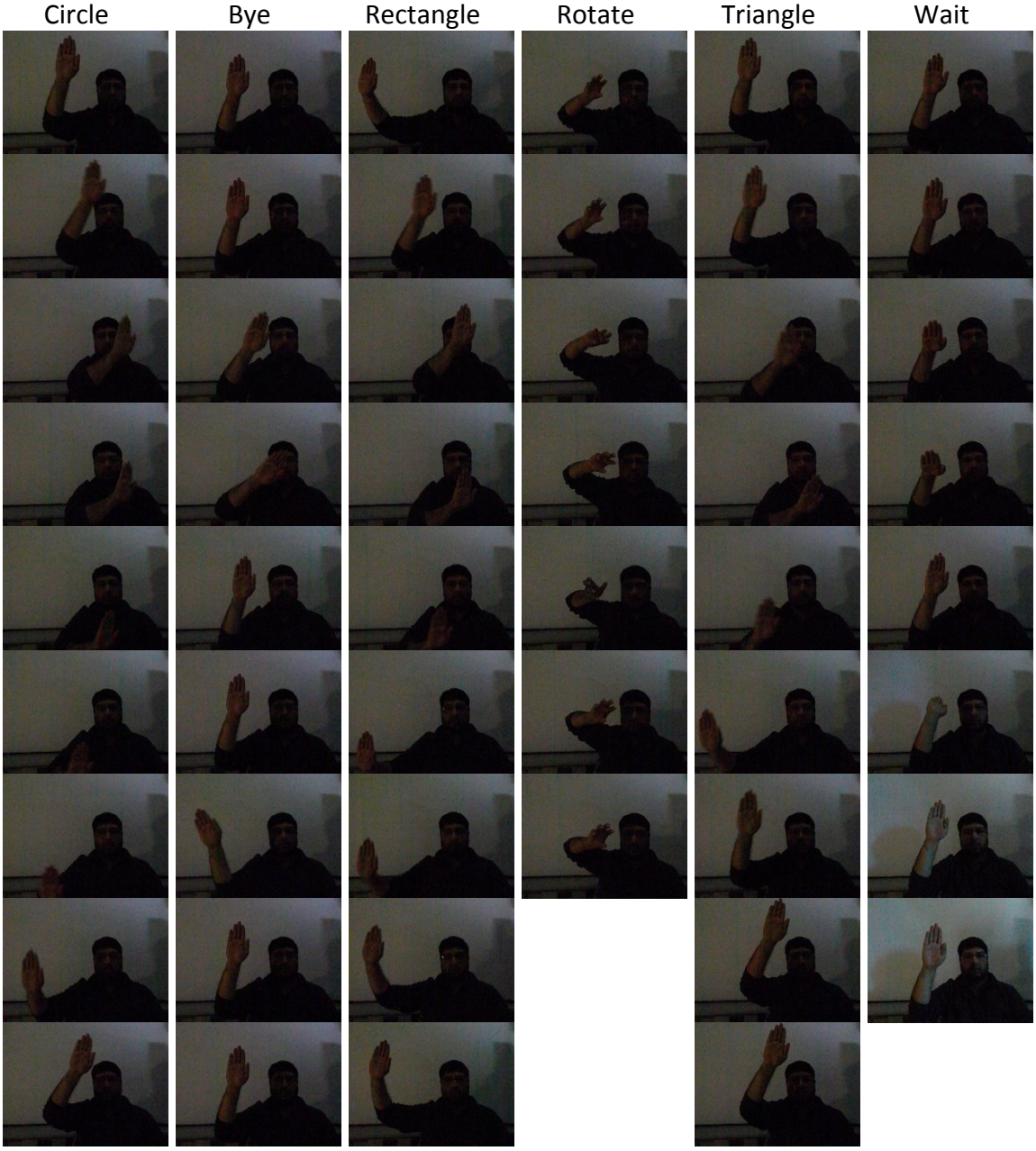


Figure 19 Straight arm, close to camera, bad light, white background

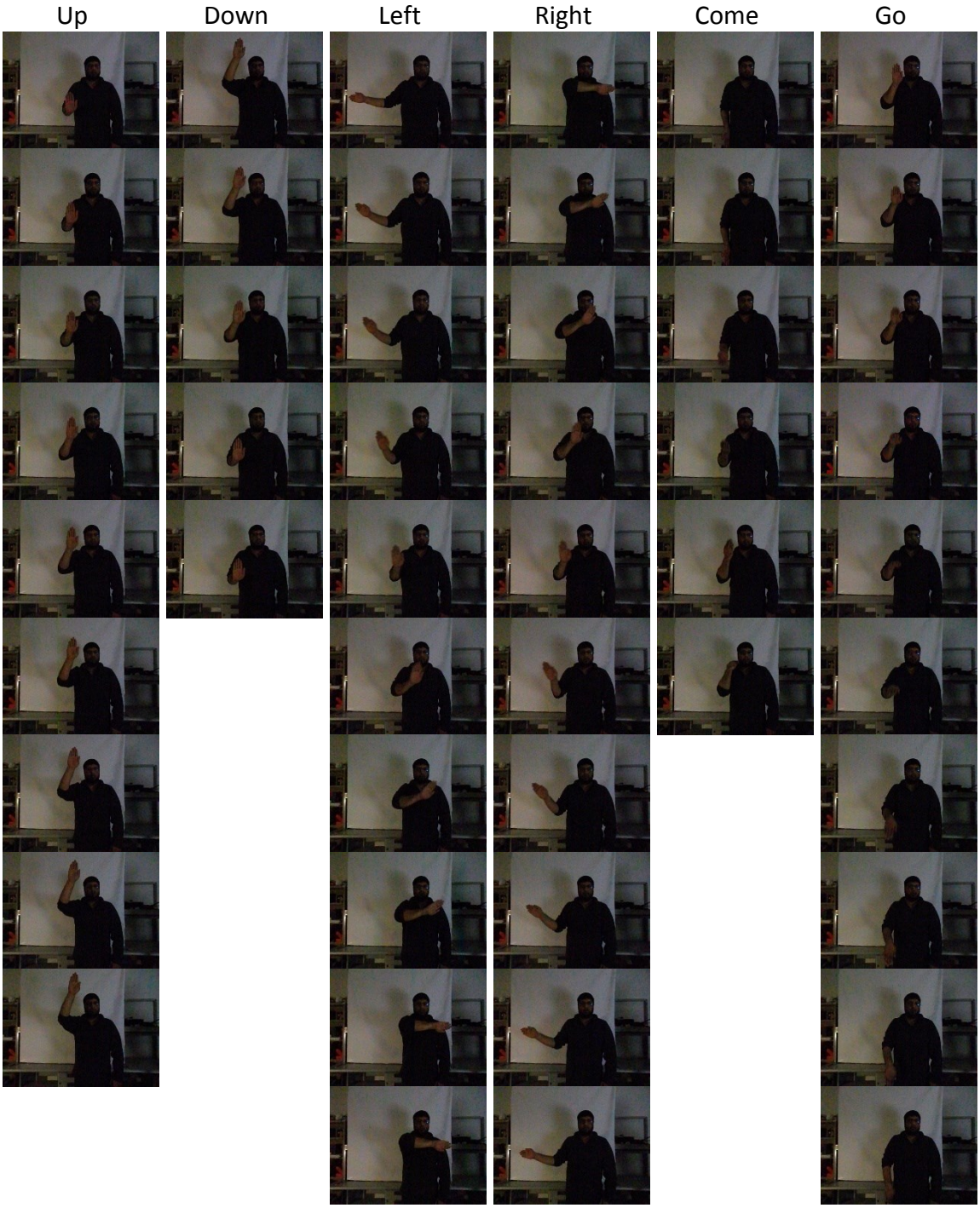


Figure 20 Straight arm, close to camera, bad light, white background

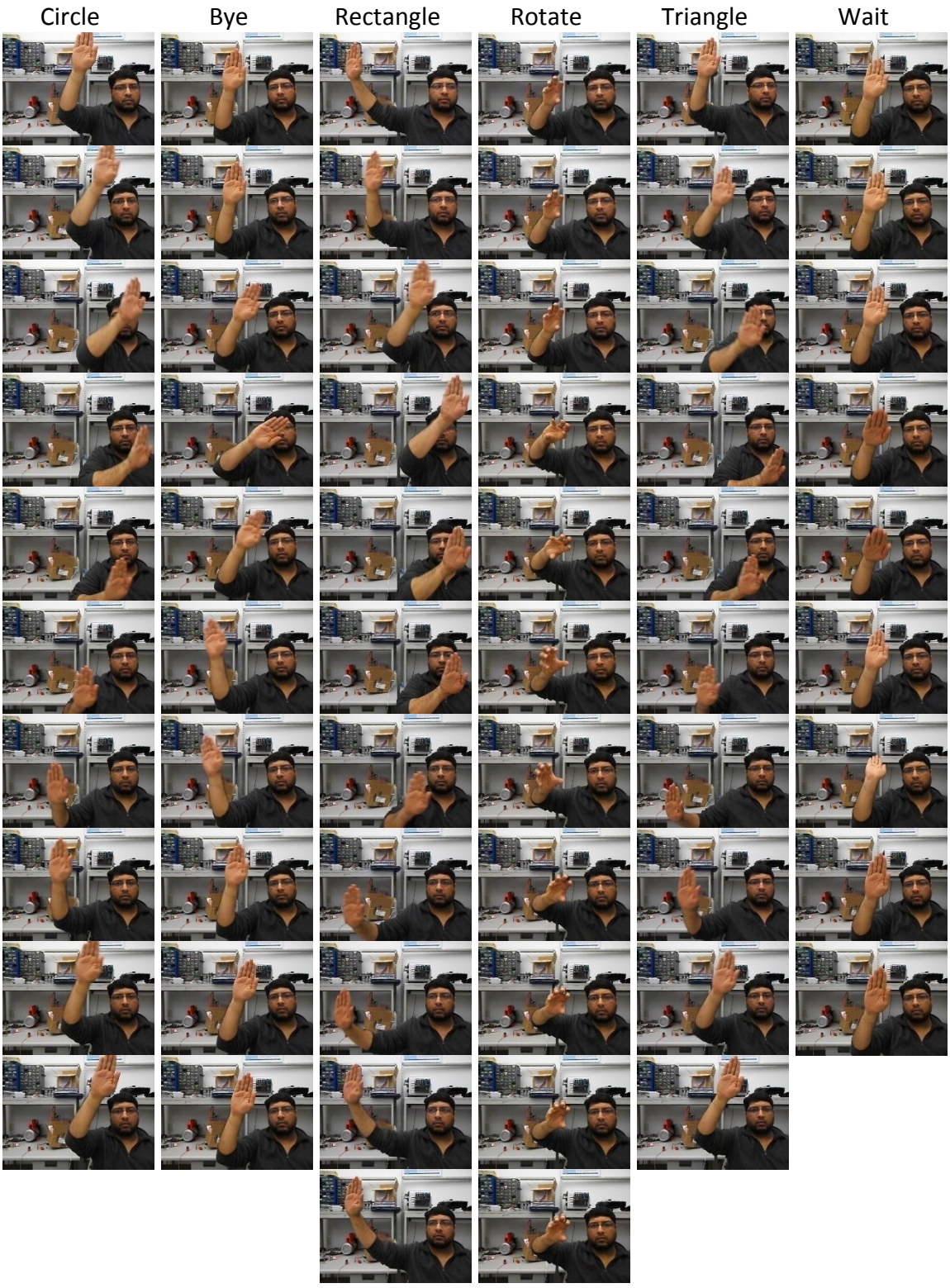


Figure 21 Straight arm, close to camera, good light, clutter background

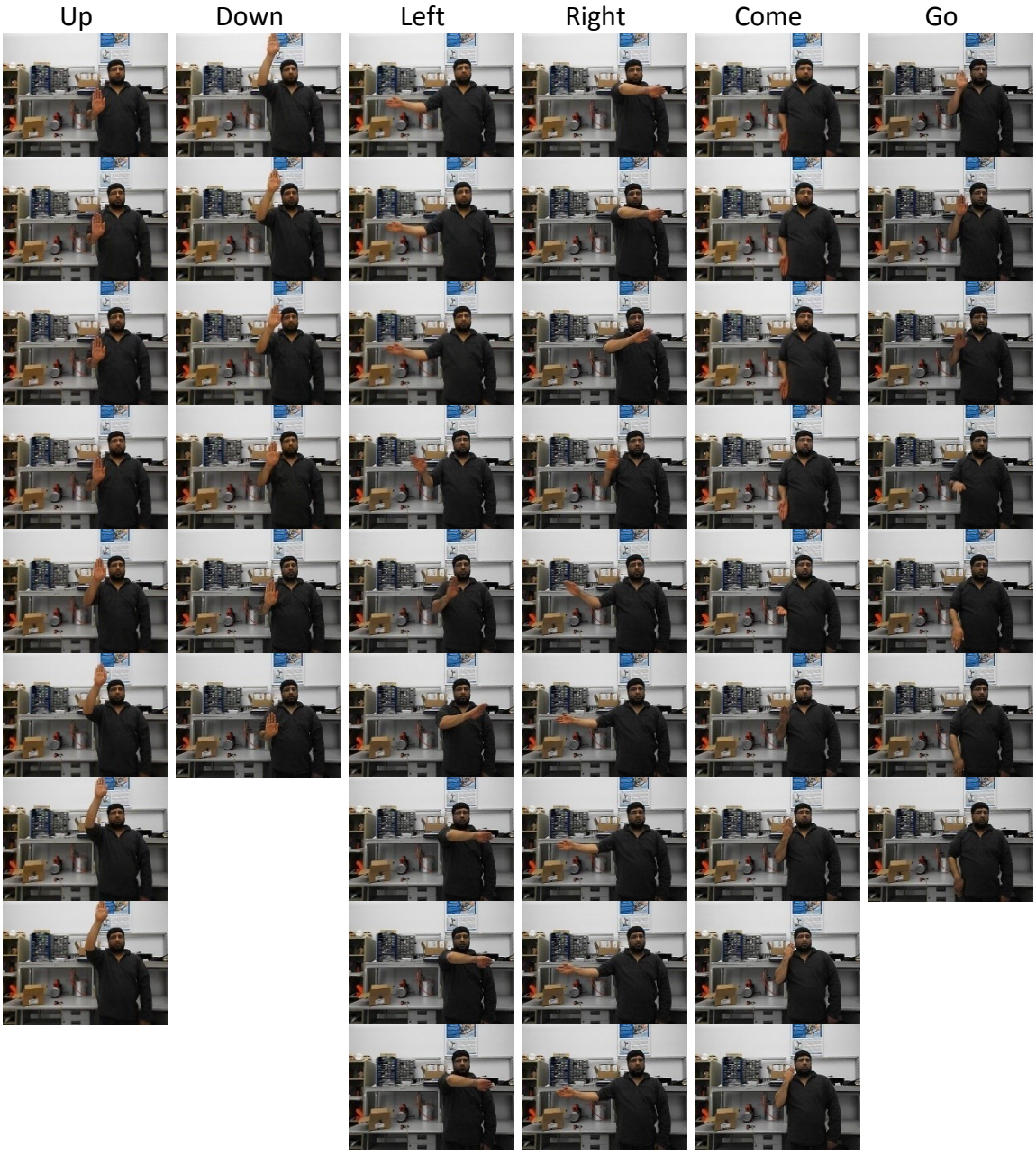


Figure 22 Straight arm, close to camera, good light, clutter background

The *left gesture* is like saying to someone by hand and forearm direction to go to left direction. It starts from positioned hand and forearm at middle-right direction. Hand is also pointing towards right direction. Than move hand horizontally from right to middle-left side completely. Hand will also point

towards left direction. This gesture consist of hand, wrist, forearm, elbow and upper arm.

The *right gesture* is also close to left gesture but in opposite direction. It starts from positioned hand and forearm at middle-left direction. Hand is also pointing towards left direction. Later move hand horizontally from left to right completely. Hand will also point towards right direction. This gesture also include hand, wrist, forearm, elbow and upper arm.

The few gestures scenarios are shown above in Figure 19 to Figure 22.

A total of 24 scenarios have been considered while generating this manual database. These include:

1. Straight Arm, Close to Camera, Good Light, White Background;
2. Straight Arm, Far from Camera, Good Light, White Background;
3. Straight Arm, Close to Camera, Bad Light, White Background;
4. Straight Arm, Far from Camera, Bad Light, White Background;
5. Straight Arm, Close to Camera, Good Light, Cluttered Background;
6. Straight Arm, Far from Camera, Good Light, Cluttered Background;
7. Straight Arm, Close to Camera, Bad Light, Cluttered Background;
8. Straight Arm, Far from Camera, Bad Light, Cluttered Background;
9. Angled Arm, Close to Camera, Good Light, White Background;
10. Angled Arm, Far from Camera, Good Light, White Background;
11. Angled Arm, Close to Camera, Bad Light, White Background;
12. Angled Arm, Far from Camera, Bad Light, White Background;
13. Angled Arm, Close to Camera, Good Light, Cluttered Background;
14. Angled Arm, Far from Camera, Good Light, Cluttered Background;
15. Angled Arm, Close to Camera, Bad Light, Cluttered Background;
16. Angled Arm, Far from Camera, Bad Light, Cluttered Background;
17. Vertical Arm, Close to Camera, Good Light, White Background;
18. Vertical Arm, Far from Camera, Good Light, White Background;
19. Vertical Arm, Close to Camera, Bad Light, White Background;
20. Vertical Arm, Far from Camera, Bad Light, White Background;
21. Vertical Arm, Close to Camera, Good Light, Cluttered Background;

22. Vertical Arm, Far from Camera, Good Light, Cluttered Background;

23. Vertical Arm, Close to Camera, Bad Light, Cluttered Background;

24. Vertical Arm, Far from Camera, Bad Light, Cluttered Background;

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.

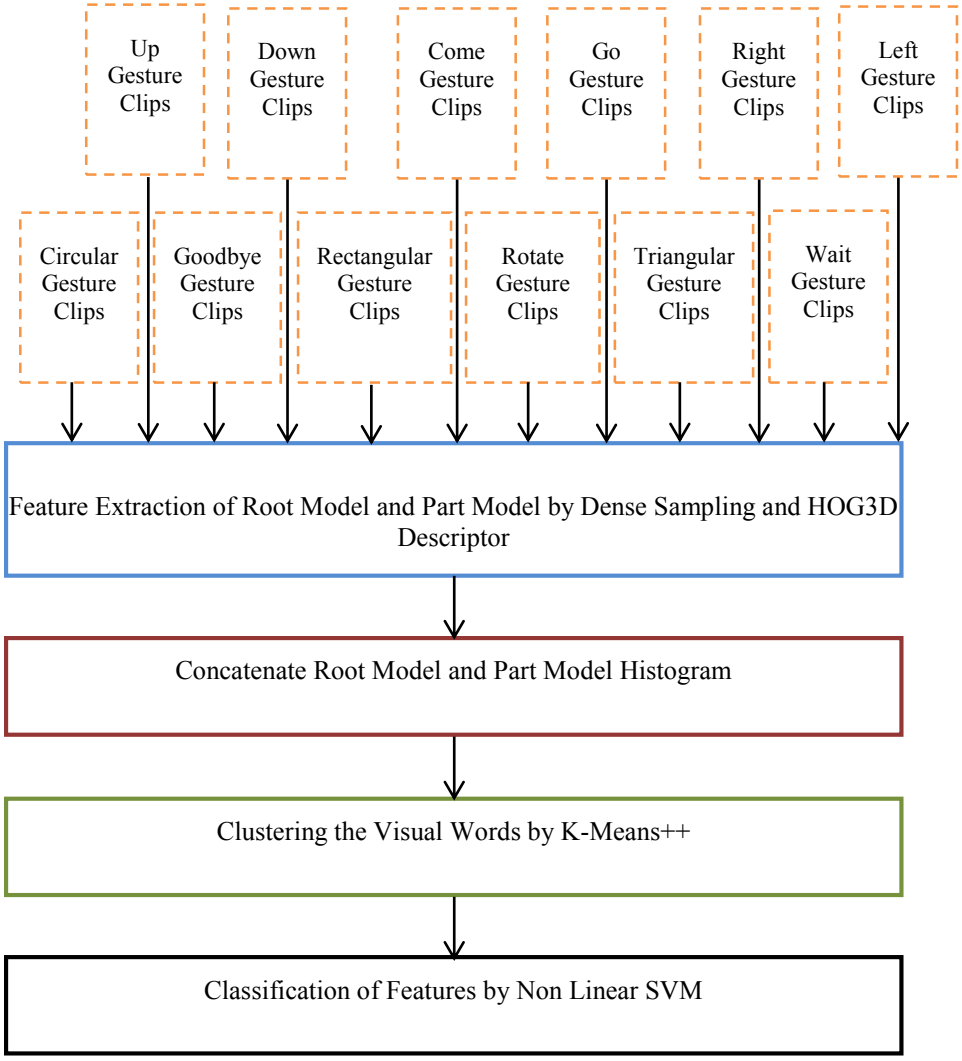


Figure 23 Overview of the training stage

In order to compensate for the inherent impossibility for a human to repeat precisely time over time the same gesture, we decided to train the IP module over a larger database of gestures recorded with 40 subjects in different background conditions. For various stages, 30 subjects were considered for the training stage and ten subjects (including the author) were specified for the testing stage. We captured all of the training clips at 640 × 480 pixels, and then reduced them to 160 × 120 pixels. This size reduction of training clips, increased the feature extraction and classification speed. Most importantly, the size reduction of training clips left no effect on the recognition of such features. The training stage model is shown in above Figure 23.

When using the BOF model to extract features from the video sequence we reduced the size of each video clip by down sampling. The dense sampling method was used to extract 3-D local ST patches from the down-sampled new videos at different scales and locations. In [94], Wang et al. demonstrated that dense sampling at regular space-time grids outperformed state-of-the-art interest-point detectors. Similar results have also been observed in [95] [109]. Compared with interest point detectors, dense sampling captures most information by sampling every pixel in each ST scale.

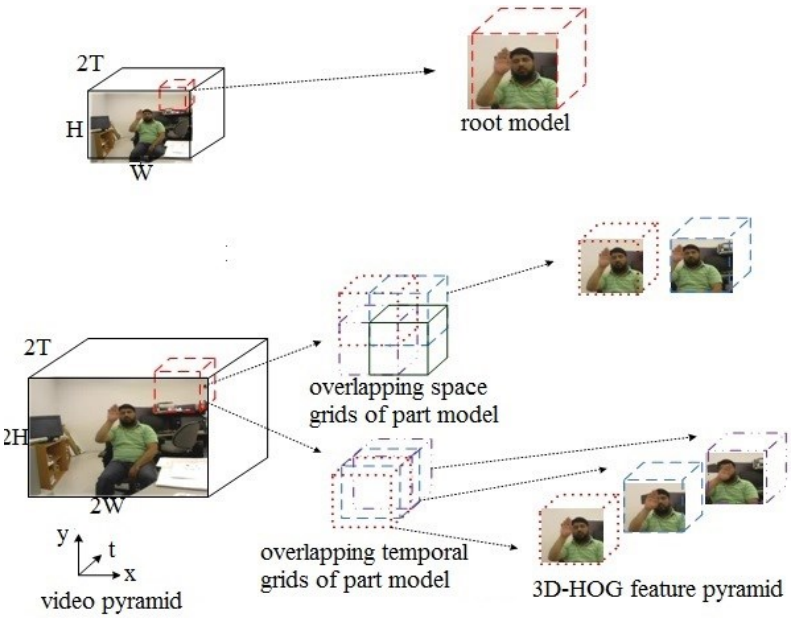


Figure 24 Feature defined by root model and overlapping grids of part model

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 Figure 24. We used a 50% overlapping ratio, as recommended in [8][12]. Later, a HOG3D descriptor [103] was used to depict the feature from both its root and part models.

Our model consist of a coarse root patch and a group of higher resolution part patches. We calculated histograms of each root and part model using HOG3D, which is based on a 3-D oriented gradient [102][140]. 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 arm gestures would be lost. We overcame this by concatenating histograms of both root and all part models. This maintained the order of dynamic arm 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. Visual vocabulary was created using dense sampling of dynamic gesture training videos; HOG3D was specifically used for vocabulary representation.

Clustering divides a group into subgroups, called clusters, so that elements in the same cluster are similar in some sense. It is implemented using an unsupervised learning algorithm and an ordinary method for statistical data analysis applied in several fields, such as machine learning, pattern recognition, image analysis, data mining, and bioinformatics. The number of the clusters (the codebook size) depends on the structure of the data. There will be a sort of compromise for how to choose vocabulary size or number of clusters. If it is too small, then each bag-of-words vector will not represent all the key points extracted from its related image. If it is too large, then there will be over fitting because of insufficient samples of the key points extracted from the training image.

We applied the k-means++ [141] method to cluster the features. This approach helped to choose random starting centers with very 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.

The support vector machine (SVM) method [142], is a new and promising classification technique. SVM has shown strong generalization ability in a number of application areas. It maps the training data into a higher dimensional feature space. A hyperplane is then constructed in this feature space that bisects the two categories and maximizes the margin of separation between itself and those points lying nearest to it. The non-linear SVM maps the input into a high-dimensional feature space by using non-linear mapping and then the linear hyperplane can be found in feature space.

We used a nonlinear SVM with RBF (radial basis function) kernel for classification and the library for SVMs [76]. Data scaling was completed on all testing and training data.

3.1.2 Overview of the Testing Stage

The testing stage used the same previously mentioned 24 scenarios that were created by the ten subjects. We generated another testing database using the same twelve 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 [8][12]. 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 arm 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 key points 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 little percentage of recognition by increasing the codebook size. Our experiments shows that increasing the size after 4000 is not resulting in a significant improvement. The increments in recognition rate and precision of results is ideal for increments between 1000 and 4000.

These calculated increments allow to obtain a 97.22% aggregated result of gesture recognition with twelve gestures and 24 scenarios, as mentioned in —Table 1. —Table 1 shows total 12 dynamic arm gestures that we consider for recognition. For each gesture, we recorded 240 video clips and —Table 1 results also show the number of incorrect recognition. Last column of table shows correct recognition percentage of dynamic gestures respectively. Each gesture has been thoroughly tested in numerous scale, rotation, illumination, and background settings.

In our experiments, we choose 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 8x8x3; number of histogram cells 2x2x2; polyhedron type dodecahedron (12); and number of parts per “root” model 2x2x2. The dimension

for “root” model is 2x2x2x12 = 96. The vector size of a feature is 96 x (1(root) + 8(parts)) = 864. We conducted experiments with different values, but we concluded that parameter values suggested by [8][12] are the best combination.

Table 1 Performance of gestures recognition for 12 gestures and 24 scenarios

Gestures	Correct recognition out of 240	Incorrect recognition out of 240	Correct recognition%
Circular	233	7	97.08%
Goodbye	234	6	97.50%
Rectangular	230	10	95.83%
Rotate	229	11	95.42%
Triangular	237	3	98.75%
Wait	235	5	97.92%
Left	233	7	97.08%
Right	232	8	96.67%
Up	236	4	98.33%
Down	238	2	99.17%
Come	231	9	96.25%
Go	232	8	96.67%
Total	**2800**	**80**	**97.22%**

3.2. Calibration of Image Processing (IP) Module

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 are 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 arm with combinations of one iteration of one gesture and a half iteration of another gesture and checked our system recognition rate 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 2. These videos were made by considering the same parameters discussed in [12]. 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 [12][142].

We checked all combinations of dynamic gestures. Few samples of video results are shown in Table 2. These results led us to the successful implementation of the grammar and formal language of our dynamic gesture language; below we mention why in more detail.

Prior to the test reported in Table 2, we had also created a test set of dynamic arm 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 in 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 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.

Table 2 Dynamic gestures with noise and their results

Dynamic gestures with noise	Checked against the gesture	Result
Half circle and complete bye	Bye	Pass
Half circle and complete bye	Circle	Fail
Complete bye and half circle	Bye	Pass
Complete bye and half circle	Circle	Fail
Half rectangle and complete triangle	Triangle	Pass
Half rectangle and complete triangle	Rectangle	Fail
Complete triangle and half rectangle	Triangle	Pass
Complete triangle and half rectangle	Rectangle	Fail
Half circle and complete rectangle	Circle	Fail
Half circle and complete rectangle	Rectangle	Pass
Complete rectangle and half circle	Circle	Fail
Complete rectangle and half circle	Rectangle	Pass

3.3. Conclusions

The human-robot communication system presented in this chapter applied a novel method to recognize dynamic arm gestures, and achieved state-of-the-art performance. The robust IP module recognizes the dynamic arm gestures, which are our "visual words", as accurately as 97.22 % by applying a novel technique for the task.

We used a database of twelve dynamic gestures to train in a robust way our IP module. A total of 24 scenarios have been considered while generating this database. These 24 scenarios are based on a combination of diverse environmental conditions which are: good light, bad light, white background, clutter background, close to camera, far from camera, straight arm, angled arm and vertical arm. Some of our gestures resemble to each other and some are very different in term of use

of the space and position of arm. Our IP module demonstrated good robustness while recognizing the set of twelve gestures.

The IP module uses a new strategy to recognize dynamic bare arm gestures from a video stream that was never applied to this domain before. This approach helps to maintain the sequence of events. Our experiments show that this combination of techniques achieves a satisfactory performance under variable scale, orientation, background, and illumination conditions.

Chapter 4. Dynamic Arm Gesture Recognition for Androids

4.1. System Architecture Overview of Inter-Robot Communication

Inter-robot dynamic arm gesture recognition is a novel development in the new e-Society paradigm. It is expected that the inter-robot and human-robot communication systems use same gestures, so both may be part of a larger common communication modality.

As discussed our DGLR system should offer a solution for this human-like android communication modality.

The arm gestures performed by android correspond to the set of human arm gestures for which the IP algorithm described in previous chapter was trained. It may be worth mentioning that the specific environmental conditions in which an android is expected to perform are actually less stringent then the much tougher conditions under which the IP module was trained and tested.

In section 4.1.1, we will briefly discuss about dynamic arm gestures performed by android and section 4.1.2 will put light on recognition detail of dynamic arm gestures perform by robot. Here in this section we will also give information of performing same gestures by human as subject in same environment to see the comparisons of performance of dynamic gesture recognition algorithm for both subjects (human and robot).

4.1.1 Robotic Arm Gesture Control

We used a first generation humanoid "Hawk" robot, H20 series, manufactured by Dr Robot Inc. [132] shown in Figure 25.

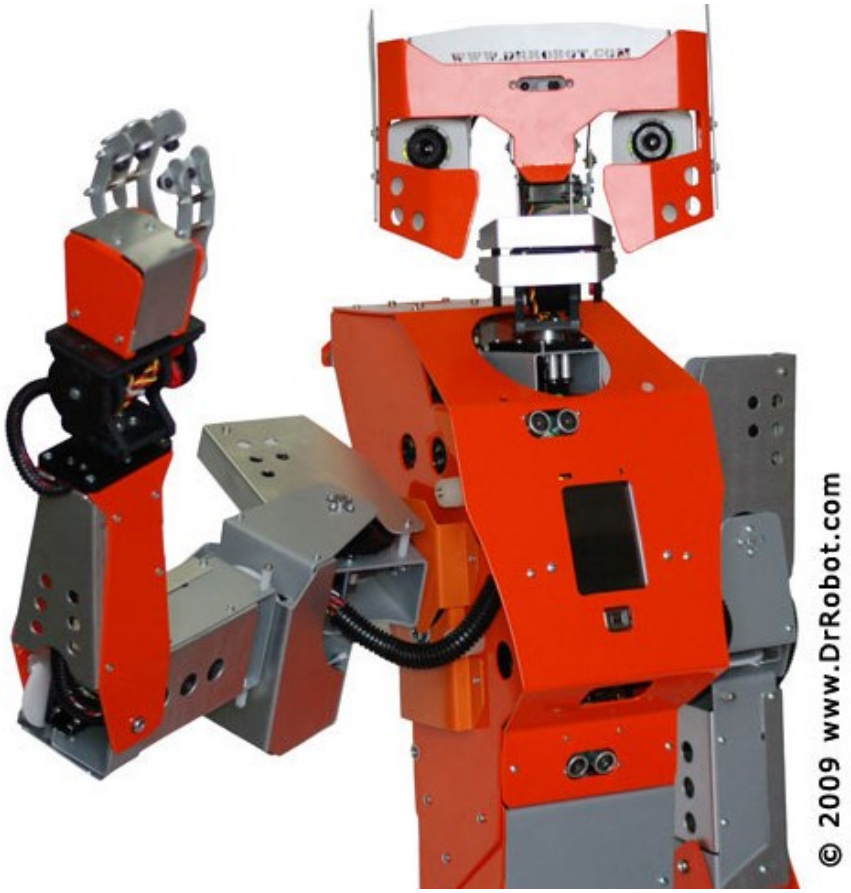


Figure 25 The arm of the "Hawk" robot produced by Dr Robot [132]

The arm of this robot is able to perform many human like gestures. The set of gestures that were programmed included "circular", "goodbye", "rectangular", "rotate", "triangular", "wait", "up", "down", "left", "right", "come" and "go". The "rectangular" and "triangular" gestures were broken down into sequential sets of discrete linear motions while the "circular", "goodbye", "rotate", "wait", "up", "down", "left", "right", "come" and "go" gestures were simulated using continuous motions.

The "rectangular" gesture consisted of four distinct linear motions of the arm beginning in the top right of the video frame and shaping a rectangle. During the entire gesture, the palm faced the camera and the fingers were extended. The "triangular" gesture differed only in shape from the rectangular gesture.

The “circular” gesture was created parametrically using continuous orthogonal sinusoidal velocities in the plane parallel to the palm. Similarly to rectangular and triangular gesture, the palm faced the camera and fingers were extended during video capture.

The “goodbye”, “wait” and “rotate” gestures were created by rotating the forearm and the palm in a continuous sinusoidal motion. The “rotate” gesture had a clasped hand while the “goodbye” and “wait” gestures had an open hand.

The “up” gesture starts with an open palm facing the camera with the fingers upwards at the altitude of the torso, and the arm moves straight up and finally stops when passed the head’s height. The “down” gesture is the same as the “up” gesture except for the arm movement which is the reverse in direction. It starts from the top and moves downwards and stops at the altitude of the torso.

The initial point of the “left” gesture is the position of the hand when the right forearm and upper arm stretch towards the right side of the body with the palm open and facing the camera and the fingers towards the right. The arm moves to the left in the same horizontal plane and rotates counter-clockwise meanwhile until it reaches the left side of the body. The position of the fingers is upwards in the middle and leftward in the end of the motion. In this transition from right to left, the palm constantly faces the camera.

The “right” gesture is the reverse of the “left” gesture. It starts from the end state of the “left” gesture and moves back its original start point. It is performed with the same arm (i.e. the right arm) and the rotation is clock-wise.

The “come” gesture starts by keeping the forearm and upper arm straight down and palm facing the camera and fingers pointing downwards. The forearm and hand move upward by bending elbow all the way up, resulting in fingers pointing upwards and the back of the hand facing the camera.

The “go” gesture starts by keeping forearm straight up with bent elbow, and palm facing the camera with fingers pointing upwards. The forearm moves downward by unfolding elbow all the way down until the forearm is straight down and the fingers point downwards and back of the hand facing the camera. Figure 26 to Figure 28 show the unified modelling language (UML) sequence diagrams

for some above mentioned dynamic gestures. Remaining UML diagrams are in Appendix A.

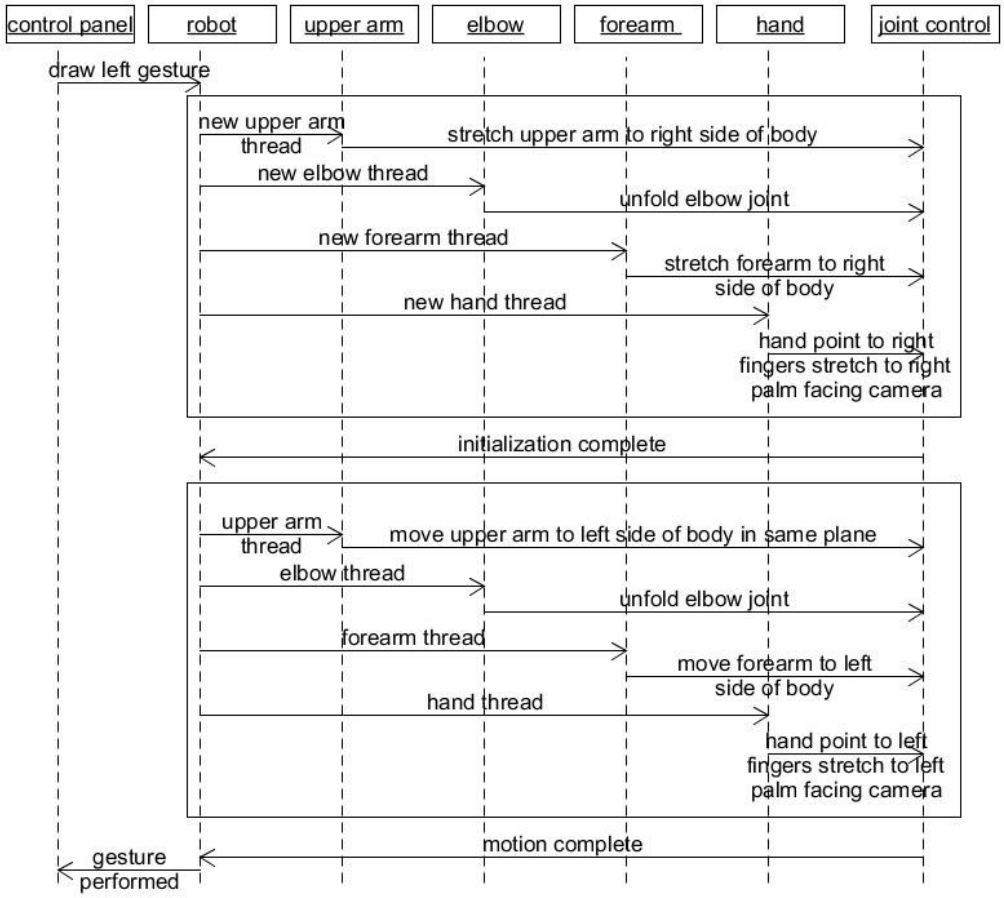


Figure 26 UML sequence diagram for “left” gesture

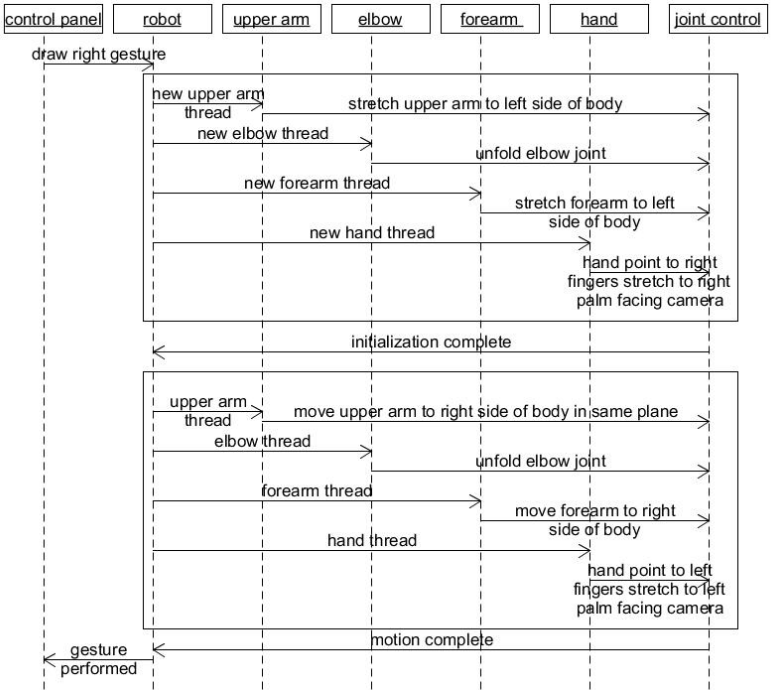


Figure 27 UML sequence diagram for “right” gesture

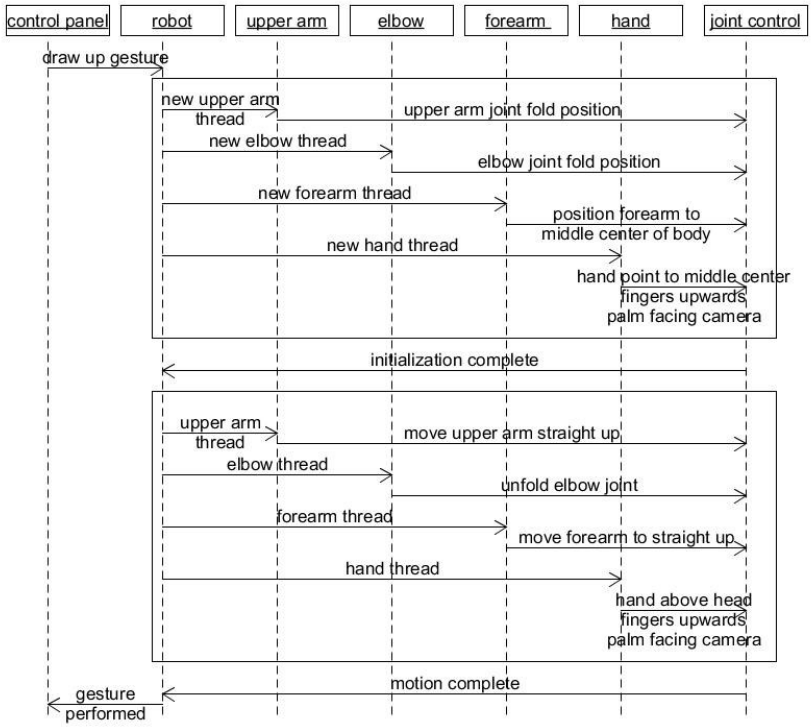


Figure 28 UML sequence diagram for “up” gesture

4.1.2 Robot Arm Gesture Recognition

When humans perform dynamic gestures, it is not practically possible for them to repeat the same gesture by using the same speed, space and arm orientation. Whereas, a robot has the ability to perform the same gesture consistently every time, with the same speed, space and orientation of arm.

As discussed in previous section, we made a new database of videos of a human as a subject and a robot, developed in our DISCOVER lab, as subject. Both types of videos (i.e. human and robot) are created with the same scenarios and conditions. We consider the same conditions as mentioned in chapter 3 for human-robot communication. These conditions are straight arm, angled arm, vertical arm, good light, bad light, close to camera, far from camera, white background and clutter background. We used 24 scenarios based on just mentioned conditions. We consider the same number of previously described dynamic gestures (12 gestures) namely: “circular”, “goodbye”, “rectangular”, “rotate”, “triangular”, “wait”, “up”, “down”, “left”, “right”, “come” and “go” gestures. A few gestures scenarios are shown in Figure 29 to Figure 32.

Positional and movement descriptions of these dynamic gestures are the same as discussed in Chapter 3 and as shown in UML sequence diagrams. All captured video clips had a resolution of 640 x 480 pixels, and were then reduced to 160 x 120 pixels. We applied our IP module to recognize the dynamic gestures of both human and robot videos. For each subject, robot and human, we recorded 288 clips for testing.

Figure 29 Straight arm, close to camera, good light, white background

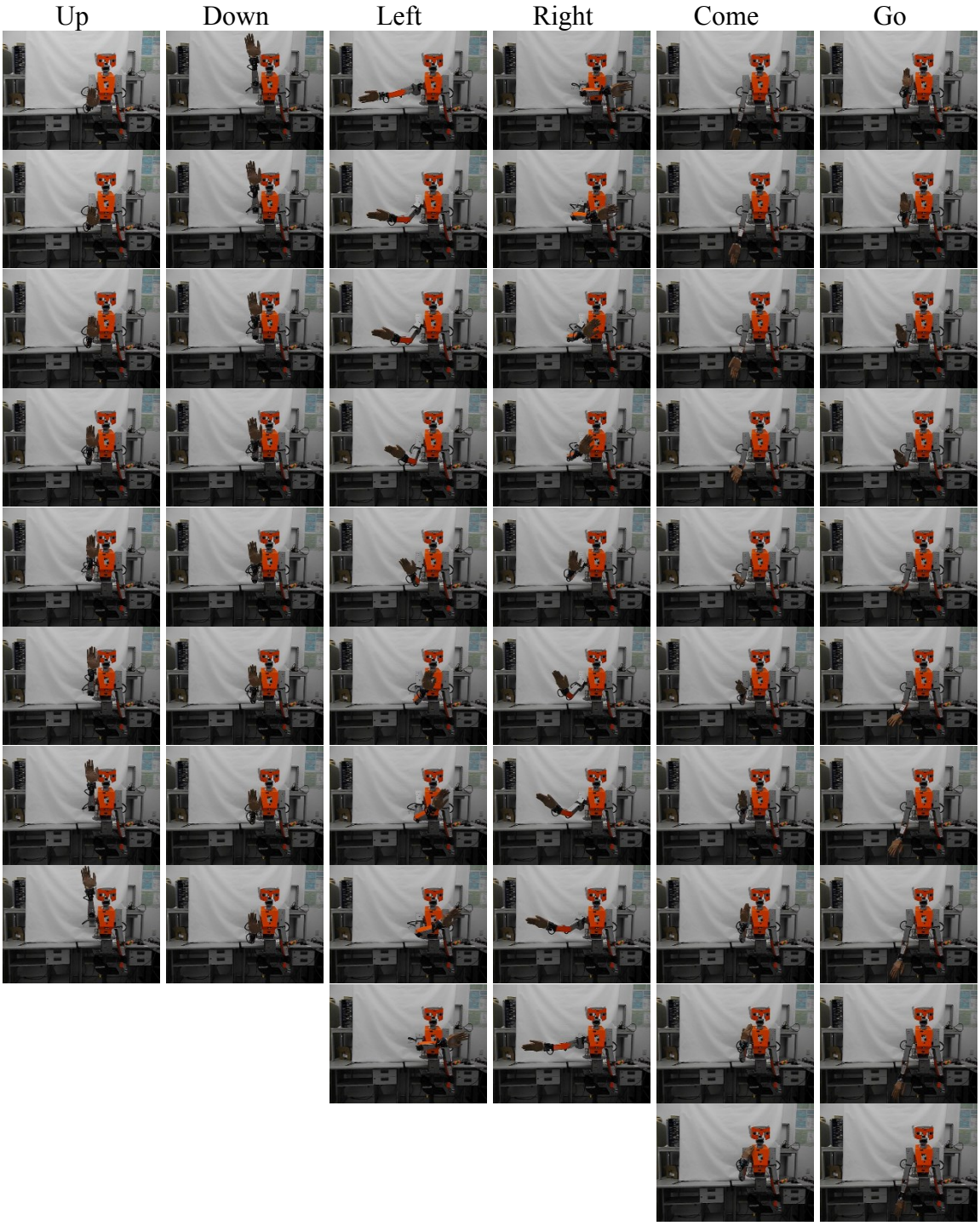


Figure 30 Straight arm, close to camera, good light, white background

Figure 31 Straight arm, close to camera, bad light, clutter background

Figure 32 Straight arm, close to camera, bad light, clutter background

The underlying building blocks for our models are local spatio-temporal volume patches, which can be extracted by dense sampling at different scales and locations. Our model consists of coarse root patch and a group of higher resolution part patches. We created the histograms of root patch and all part patches, concatenated all patches to create histogram representation of a local ST feature. Both the coarse root patch and higher resolution part patches are described by HOG3D descriptor. In our experiment, neighbouring sub-patches are 50% overlapped. We concatenated histograms of both root and all part models. This maintained the order of dynamic arm gestures. Local ST Features have been represented by concatenated histograms to keep the order of the event. These extracted features are quantized in visual words. Frequencies of these visual words were measured for classification. Visual vocabulary was created using dense sampling of dynamic gesture training videos.

We applied the k-means++ [141] method to cluster the features. This approach helped to choose random starting centers with very 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 the BOF and nonlinear SVM with RBF (radial basis function) kernel for classification. We conducted experiments with 4000 visual words size of codebook. As our tests shown in IP module that these number of visual words give good results as compare with 1000, 2000, 3000 and 5000 till 10000 visual words size of codebook. Data scaling was completed on all testing data.
Table 3 shows the results of our new database of dynamic arm gestures.

Table 3 Performance of gesture recognition of human and robot arm

Dynamic gesture subject	Correct recognition out of 288	Incorrect recognition out of 288	Correct recognition %
Human	280	8	**97.22%**
Android	269	19	**93.40%**

Results shown in Table 3 are very satisfactory for inter-robot communication by using our proposed IP module. Robot arm gestures resulted less percentage of recognition rate as compare to human arm gestures. It is because of experimental robot restricted DOF as compare to human arm.

Our research contributes to a larger effort to develop efficient practical solutions allowing humanoid robots to understand the gestures performed by humans as well as by other humanoid-robots, and also to provide the humanoid-robots with the ability to communicate in a human-like way with both humans and humanoid robots.

This will contribute to improve the teamwork quality of humanoid robots and humans in various domains like industry, public sector and consumer market and enable a new era for robots to serve in human society. Making possible for humanoid robots to understand human dynamic gestures will help humans to enjoy an independent life.

Elder care is one of big markets for humanoid robots as our society is facing a big challenge in the long term elder care sector due to lack of trained health care staff. At this stage of research, it still may look hard to accept that robots can replace human care givers.

In the same way humanoid robots able to work together with humans could contribute in the challenging environment of manufacturing industry or on mission-critical applications such as cleaning ecological disaster areas.

4.2. Conclusions

This chapter describes an original android arm dynamic gesture recognition system which we developed for inter-robot communication.

The experimental testbed allows to generate consistently repeatable robot arm gestures. We created a new database of dynamic gestures performed by the android. Our experiments showed that the recognition rate for the android subject was 93.40%, slightly less than the 97.22% for human subjects. In our opinion this is due to more mechanical constraints and less DOF of the android's arm.

Chapter 5. Dynamic Gesture Language Recognition Using Formal Grammars

Arm gesture recognition is one of the most commonly used communication modality in human-to-human and in human–robot communication. The dynamic arm gesture communication is a more powerful mode of communication with robots, as compared with static hand gestures. This is due to the fact that arm in the dynamic mode is allowed to move in any direction, and bend toward any angle in all accessible coordinates. In contrast, static hand gesture communication is constrained to a more limited set of possible postures.

Currently, dynamic arm gestures have been adopted by a number of applications, including smart home, video surveillance, and long-term healthcare environment applications. All of these applications require maximum recognition rate and maximum performance against the time and a cluttered background. We discussed in Chapter 4 about dynamic arm gesture recognition performed by humanoid robot which is developed in our laboratory at the University of Ottawa.

The Dynamic Gesture Language Recognition (DGLR) system which we propose comprises of two subsystems: an Image Processing (IP) module, and a Linguistic Recognition System (LRS) module. LRS has further two subsystems: derivation grammar which is based on Linear Formal Grammar (LFG) module, to derive the valid sequence of dynamic gestures and other is parsing grammar which is based on Stochastic Linear Formal Grammar (SLFG) module, to predict the unrecognizable gesture by IP module. This chapter discusses our DGLR system and covers the details of the LRS module and completes our introduction to the DGLR system.

The LRS module analyzes the sentences (i.e., sequences of gestures) of the dynamic arm gesture language and determines whether or not they are syntactically valid. Therefore, the DGLR system is not only able to rule out ungrammatical

sentences, but it can also make predictions about missing gestures, which, in turn, can increase the accuracy of our recognition task.

Our DGLR system has a number of advantages over the previous systems. First, it has a modular structure (consisting of two modules IP and LRS) which allows for an override mechanism for misses in classification. Therefore, when there is a miss, there is still hope for that data point to be reclassified correctly. Second, older methods, based on tracking of hand, tracking of fingers, recognition of hand, recognition of fingers, etc., limit the performance to that particular part of the body, whereas the 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. Our choice of the local part model enables DGLR to recognize bare arm dynamic gesture recognition. Finally, our method also benefits from a LRS module to process a higher level regularity (syntactic constraints) in the dynamic gesture language (DGL), 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.

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 gesture language, dynamic gestures are the words. Sequences of these dynamic gestures are sentences of the language. The ultimate object of the derivation 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.

5.1. Linguistic Recognition System

During the phase of any type of communication, 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:

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

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

a) We first abstracted from the process of deriving dynamic gestures to derive sequences of dynamic gestures. Ultimately we defined a formal language to represent these sequences.
b) We developed formal grammar to accept valid sequences of the formal language and to reject invalid ones.
c) 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 arm gestures onto the corresponding symbols as shown in Table 4, 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.

Table 4 Dynamic arm gestures and the corresponding commands and symbols of the formal language

Dynamic arm 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
Up	Look up	f
Down	Look down	g
Left	Move right arm toward left	h
Right	Move right arm toward right	i
Come	Raise right arm	j
Go	Lower right arm	k

The commands and detail of their corresponding actions are as follows:

1) Wake up: the 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.
7) Look up: the robot raise his head upward
8) Look down: the robot lower his head downward

9) Move right arm to left: the robot moves his right arm to the right side of body in full stretch.
10) Move right arm to right: the robot moves his right arm to the left side of body with full stretch.
11) Raise right arm: the robot raise his right arm above the head
12) Lower right arm: the robot lower down his right arm from head to middle of body.

In next sections, we will see what we mean by a formal language, formal grammar and how they help to construct a successful human-robot and inter-robot communication.

5.1.1 Definitions and Chomsky Hierarchy of Formal Languages

There are some definitions which are worth mentioning here for better understanding of the upcoming discussions. These are:

Derivation: The process of recursive generation of sequences from a grammar.

Parsing: Finding a valid derivation using automaton.

Parse tree: The alignment of the grammar to a sequence.

As shown in Figure 13, regular grammars are the most restricted and consequently least inclusive of the possible structures. This means if they are able to perform the sequential classification task they will be the most efficient system as compared to the more complex grammars. That is why we assume that our DGL is regular (i.e. linear) and we will devise a stochastic linear formal grammar to predict the missing dynamic gesture prediction task. Theoretically there is only one drawback that the chosen system may encounter and that is, it would not be as efficient for the more complex sequence.

Figure 33 and Figure 34 compare sentences of a context-sensitive language and a regular language. It should be noted that a context-sensitive language contains every sentence of a regular language (that is produced by a linear formal grammar) but on the contrary a linear formal grammar cannot produce all the sentences that can be produced by a context sensitive grammar. Figure 34 is a result

of a context sensitive grammar that can produce strings such as $a^n b^n c^n$ whereas but a linear grammar cannot.

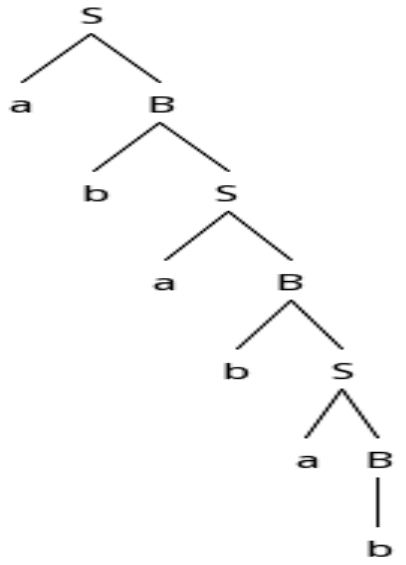


Figure 33 An example tree diagram of a sentence of a regular language, adopted from [146].

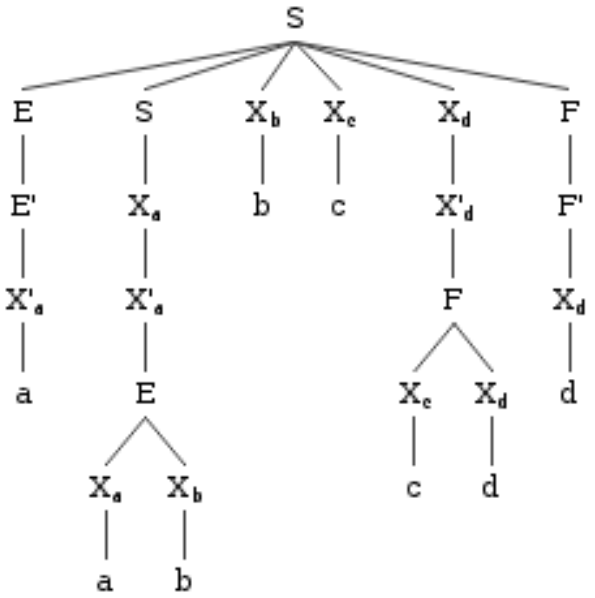


Figure 34 An example tree diagram of a sentence of a context-sensitive language, adopted from [146].

5.1.2 Derivation Grammar: Linear Formal Grammar (LFG)

A chain of command 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 robot to be turned on right after it is turned on. Be as it may, two or more consecutive "turn on" commands are invalid. This notion can be fully captured by our formal grammar called G1.

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" [145], which is defined as $G = (V_N, V_T, P, S)$, where V_N, V_T, and P are finite sets, non-empty 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 non-terminal 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. V* is the set of all finite sequences 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 comprised of the sequences that our dynamic arm gestures were previously mapped onto:

V_T = {s, a, b, c, d, e, f, g, h, i, j, k}

Table 5 Production rules of our formal language

Production rules				
S→sA	C→cD	E→eJ	H→hD	J→jI
S→sB	C→cE	F→f	H→hE	J→jK
S→sC	C→cF	F→fA	H→hF	K→k
S→sD	C→cG	F→fB	H→hG	K→kA
S→sE	C→cH	F→fC	H→hI	K→kB
S→sF	C→cI	F→fD	H→hJ	K→kC
S→sG	C→cJ	F→fE	H→hK	K→kD
S→sH	C→cK	F→fG	I→i	K→kE
S→sI	D→d	F→fH	I→iA	K→kF
S→sJ	D→dA	F→fI	I→iB	K→kG
S→sK	D→dB	F→fJ	I→iC	K→kH
A→a	D→dC	F→fK	I→iD	K→kI
A→aB	D→dE	G→g	I→iE	K→kJ
A→aC	D→dF	G→gA	I→iF	
A→aD	D→dG	G→gB	I→iG	
A→aE	D→dH	G→gC	I→iH	
A→aF	D→dI	G→gD	I→iJ	
A→aG	D→dK	G→gE	I→iK	
A→aH	E→e	G→gF	J→j	
A→aI	E→eA	G→gH	J→jA	
A→aJ	E→eB	G→gI	J→jB	
A→aK	E→eC	G→gJ	J→jC	
B→b	E→eD	G→gK	J→jD	
B→bS	E→eF	H→h	J→jE	
C→c	E→eG	H→hA	J→jF	
C→cA	E→eH	H→hB	J→jG	
C→cB	E→eI	H→hC	J→jH	

Our non-terminal vocabulary V_N is as follows:

V_N = {S, A, B, C, D, E, F, G, H, I, J, K}

G1 production rules *P* are in Table 5.

Our linear grammar G_1 distinguishes between invalid sequences of gestures and valid ones; only then our system goes on and corrects the invalid ones. Alternatively put, our system knows what an invalid sequence of gestures is, and then it tries to predict the correct sequence intended 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 by [146]. Hence, they are most efficient in terms of processing and computational cost. They are equivalent to finite state machines. G1 captures linear context surrounding each gesture, or what validly precedes or follows each gesture.

Tree Structures of Sequences of Gestures of G1

In this section we present instances of a few sentences (i.e. sequence of dynamic gestures) derived by G1. G1 is a right linear grammar that is the strings it creates starts from the left side and stretches all the way to the right. Such may be observed in all the tree diagrams sketched in this section as the trees start from the top-left corner and expand to the right (more specifically bottom-right corner). Figure 35 depicts a derivation tree of a long sequence of dynamic gestures of our dataset (to be discussed in the next section). The sequence of gestures is parsed according to the rules of G1 as shown previously in Table 5. Figure 36 shows a medium size sequence of gestures derivation tree in our dataset and Figure 37 displays the derivation pertinent to a short sequence of gestures in the dataset. All these sequences are valid and consequently have been parsed by G1. But what if we have an invalid sequence of gestures outside of our data set? Let's have an invalid sequence of gestures from outside the dataset: seabcda**a**. Consider deriving the last chunk of this sequence that is "daa". Now if we parse from left to right there are rules in G1 to parse "da" namely D→dA and A→a operating successively,

however there is no combination of rules that can parse “aa”. It is because the sequence “aa” is not allowed by the rules of grammar. Pragmatically “aa” leads to the commands “Open both hands” and then right away another “Open both hands”. This chain of commands is not physically possible and is prohibited by the grammar (i.e. G1). As a result Figure 38 shows the parse of the sequence of gestures up to the point that it is valid. The parse stops when the sequence becomes invalid by violating G1 rules.

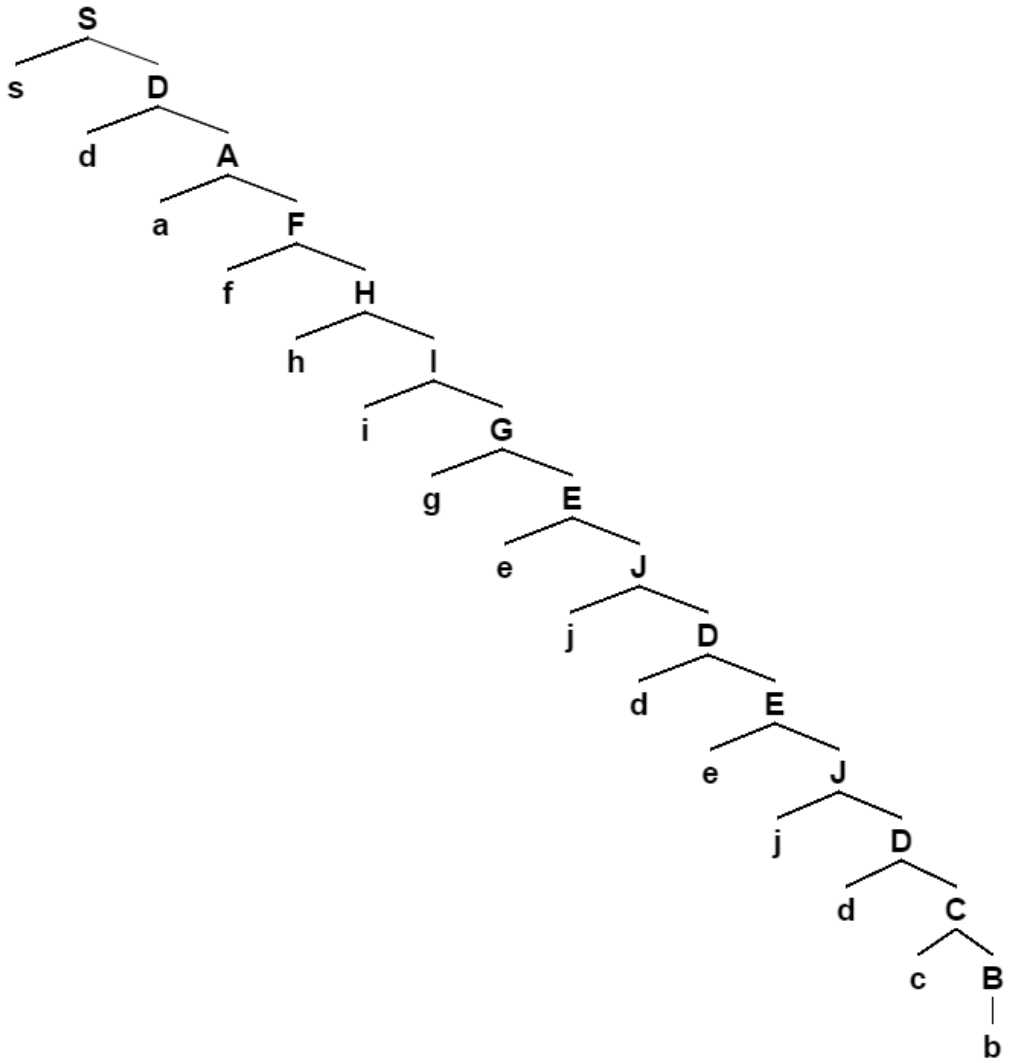


Figure 35 An instance of a long sequence of gestures derived graphically by G1: as may be observed G1 leads to a binary tree structure.

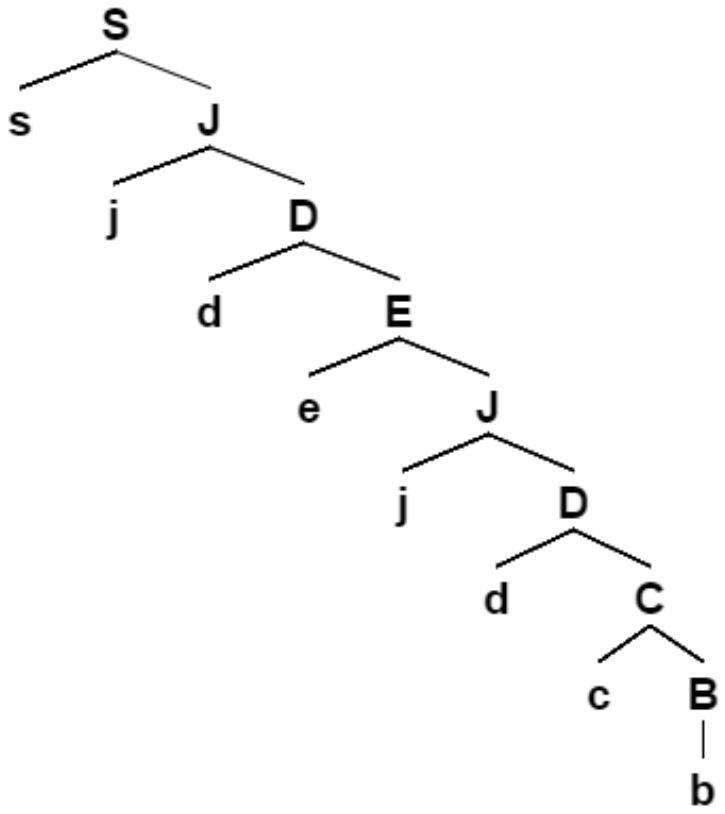


Figure 36 An instance of a medium-length sequence of gestures derived graphically by G1

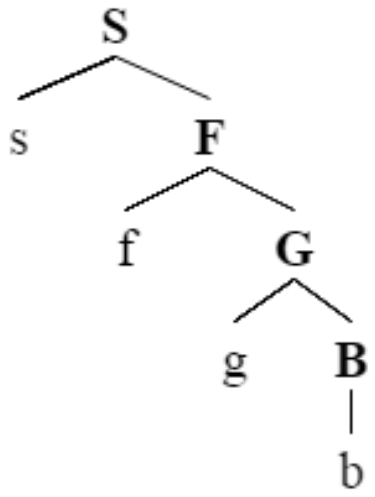


Figure 37 An instance of a short sequence of gestures derived graphically by G1

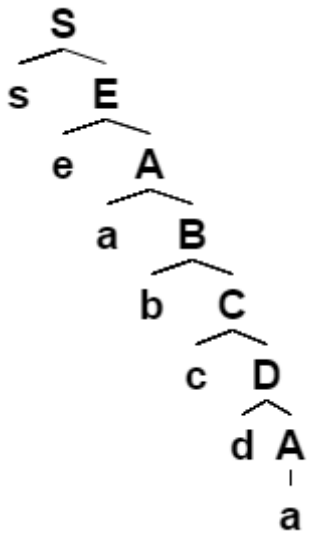


Figure 38 An instance of a sequence of gestures derived graphically by G1 in an intentional effort to violate G1. The last symbol (i.e. **a**) of the sequence seabcda**a** is invalid and will stop the derivation.

We developed software named Grammar Interface (GI) to create a language database to train the SLFG module. GI guarantees that only valid sequences of gestures find their way into the training set. Figure 39 displays GI's interface when it rejects to accept a gesture into its dialogue box. It rejects the gesture since the resulting input of gesture sequence is not allowed by the rules of G1. Using Table 4, we can translate the sequence of gestures into the formal language. The input gestures shown under the gesture display box in Figure 39 are: Wake-up raise-both-arms open-both-hands close-both-hands open-both-hands. And when the next gesture is introduced as open-both-hands the rejection message will pop up. The reason is that the sequence s d a c **a a** cannot be derive by the rules of G1 (previously shown in Table 5), and therefore is not a valid sequence of the dynamic gesture language.

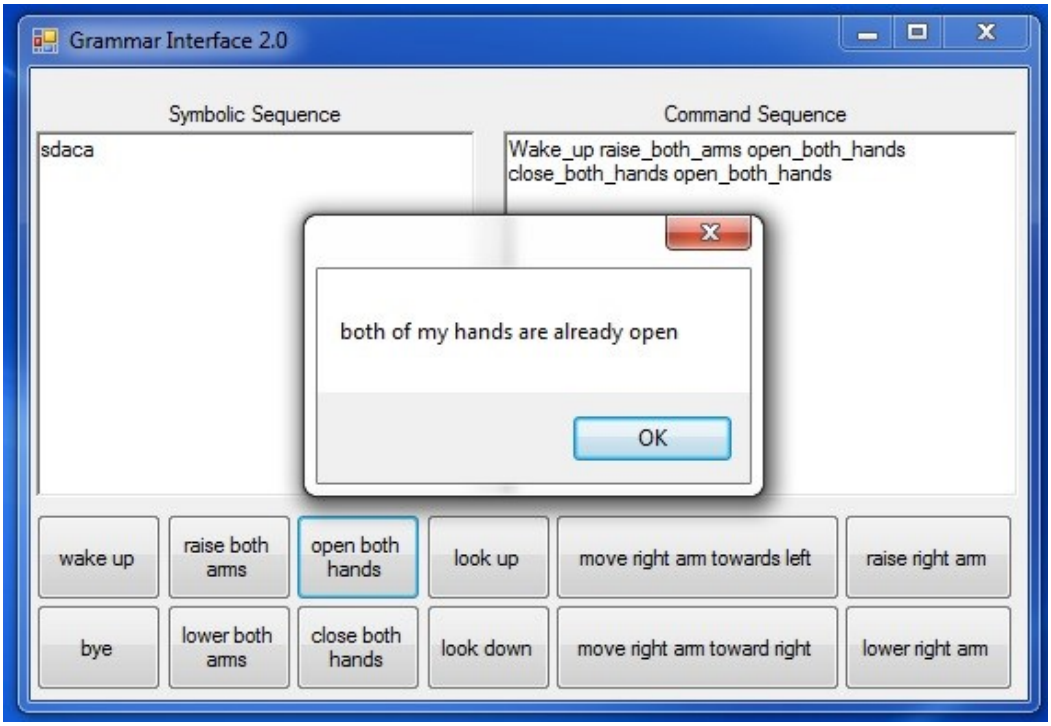


Figure 39 Grammar Interface (GI) to train the SLFG module

With GI we created two datasets for our experiments. The first dataset is made of 100 sequences of gestures of our formal language, with an average length of 10 gestures. The longest sequence of gestures in the dataset is comprised of 28 gestures, and the shortest is two gestures long. We created a second dataset by adding another 100 sequences of gestures. The average sequence of gestures length parameter was kept constant across the two datasets.

Following is the subset of our training dataset of 200 sequences of gestures. It comprises of 10 sequences made by G1 interface by following production rules. All these are valid sequences. Complete training dataset is present at end in Appendix B. It is worth mentioning that command in the following dataset is with respect to the mapping of symbols and commands shown in Table 4.

```
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
```

Wake_up move_right_arm_toward_left move_right_arm_toward_right bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up move_right_arm_toward_left open_both_hands raise_right_arm raise_both_arms bye
Wake_up raise_both_arms open_both_hands move_right_arm_toward_left bye

5.1.3 Parsing Grammar: Stochastic Grammar to Predict Non-Recognized Dynamic Gestures

We define our Stochastic Linear Formal Grammar (SLFG) as a 4 tuple (V_N, V_T, P, S) where V_N, V_T, and P are finite sets, non-empty [145]. Let $V_T \cup V_N = V$, then:

1. V_T is our terminal vocabulary
2. V_N is our non-terminal vocabulary
3. P comprises ordered groups of size three, that is (A_i, β_j, p_{ij}) where $A_i \in V_N$ and $\beta_j \in V^+$, and p_{ij} is a normalized real number showing the probability of A_i rewriting as β_j. Formally:

i. $p_{ij} = p(A_i \rightarrow \beta_j)$

We use SLFG to classify noisy input which cannot be classified by the IP module. The idea is to discover the patterns in which the commands occur in and use that knowledge to predict noisy input. Let us define "history" as the length of the string of commands that we take into account when trying to discover these patterns. By this definition, simple stochastic formal grammars use limited history. To elaborate take a look at items in 4 below:

4. Here are instantiations of (i) above:
   a. $p_1 = p(A \rightarrow aB) = 0.2$
   b. $p_2 = p(A \rightarrow aC) = 0.3$
   c. $p_3 = p(A \rightarrow aD) = 0.1$
   d. $p_4 = p(A \rightarrow aE) = 0.15$

e. $p_5 = p(A \rightarrow aD) = 0.25$

Based on the information provided by items in (4) above p_2 is our best prediction. That is if we constantly keep using this rule whenever A is confronted we will classify correctly 30% of the time. But if we widen our scope we may notice that whenever the rule S→ sA occurs we have the below probabilities:

5. Instantiations of (i) knowing that S→ sA immediately precedes.
   a. $p_1 = p(A \rightarrow aB \mid S \rightarrow sA) = 0$
   b. $p_2 = p(A \rightarrow aC \mid S \rightarrow sA) = 0.02$
   c. $p_3 = p(A \rightarrow aD \mid S \rightarrow sA) = 0$
   d. $p_4 = p(A \rightarrow aE \mid S \rightarrow sA) = 0.9$
   e. $p_5 = p(A \rightarrow aD \mid S \rightarrow sA) = 0.08$

Then p_4 (and not p_2) suddenly has by far the highest probability considering the extra information accounted for by use of a greater history. Rules in 4 have a history of size 1, it means the only information we have is the non-terminal A when predicting the next symbol. Below (6) is used to train an SLFG with history of size 1:

$$6. p(C \rightarrow cD) = \frac{N(C \rightarrow cD)}{N(C)}$$

In the equation above $p(C \rightarrow cD)$ is the probability of the non-terminal C to rewrite as cD, in other words it is the probability of the rule $C \rightarrow cD$. In addition, $N(C \rightarrow cD)$ is the number of times the rule $C \rightarrow cD$ is occurred in our training set (i.e. the dataset). $N(C)$ is the number of times the term C has occurred in our training data, rewriting as anything (either cD or any other sequence). However we would like use a more accurate model that uses more information to be used in classification, like (5) above that uses a history of size 2 (it has information that non-terminals S and A precede immediately . If we let Φ be any rule of form $A_i \rightarrow \beta_j$, from (5) and (6) we can derive (7):

$$7. \ p(C \rightarrow cD \mid \Phi) = \frac{N(C \rightarrow cD \mid \Phi)}{N(C \mid \Phi)}$$

$p(C \rightarrow cD \mid \Phi)$ is the probability of C rewriting as cD, given that Φ precedes immediately. $N(C \rightarrow cD \mid \Phi)$ is the count of $C \rightarrow cD$ following immediately the occurrence of Φ. And finally $N(C \mid \Phi)$ is the number of times C follows Φ immediately. We use equation 7 that we just derived to train our SLFG with history of size 2.

For each of the two datasets our SLFG computes the likelihood of a sequence amongst all of the possible sequences (i.e. patterns) pertaining to a missing/invalid/unrecognized gesture.

5.2. Results

We used ten-fold cross validation (Ten-fold CV) to test the performance of the SLFG module.

Ten-fold Cross Validation

In the ten-fold CV method the data which consists of 100 sequences of dynamic gestures was divided into ten equal and non-overlapping 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. ten times). The fold boundary parameters were set by the number of sequences. That is, for each training and testing pair, the data set was split into 90 sequences of dynamic gestures for training and 10 remaining separate sequences for testing. Figure 40 presents the results of the tenfold CV with 100 sentences (i.e. sequences of dynamic gestures) dataset.

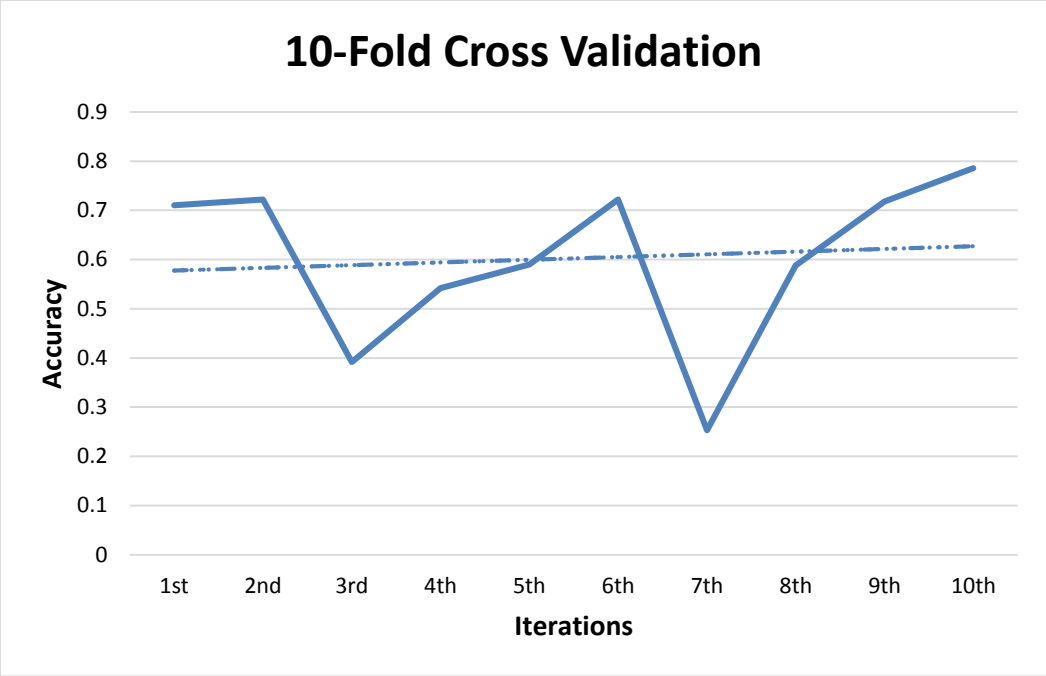


Figure 40 Tenfold CV graph with 100 sentences dataset

Table 6 below shows the results for each iteration of the 10-fold cross validation, as well as the aggregate accuracy of the 10-fold cross validation iterations with 100 sentences of dataset.

Table 6 Results of the Tenfold CV with 100 sentences dataset

Iteration	Accuracy
1	0.710
2	0.722
3	0.391
4	0.542
5	0.590
6	0.722
7	0.253
8	0.588
9	0.718
10	0.786
Aggregate CV Accuracy	**0.602**

We repeated above mentioned tenfold cross validation test with another dataset. In this setup, dataset consists of 200 sentences (i.e. sequences of dynamic gestures) was divided into ten equal and non-overlapping 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. ten 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 180 sentences (i.e. sequences of gestures) for training and 20 remaining separate sequences for testing. Figure 41 presents the results of the tenfold CV with 200 sentences dataset.

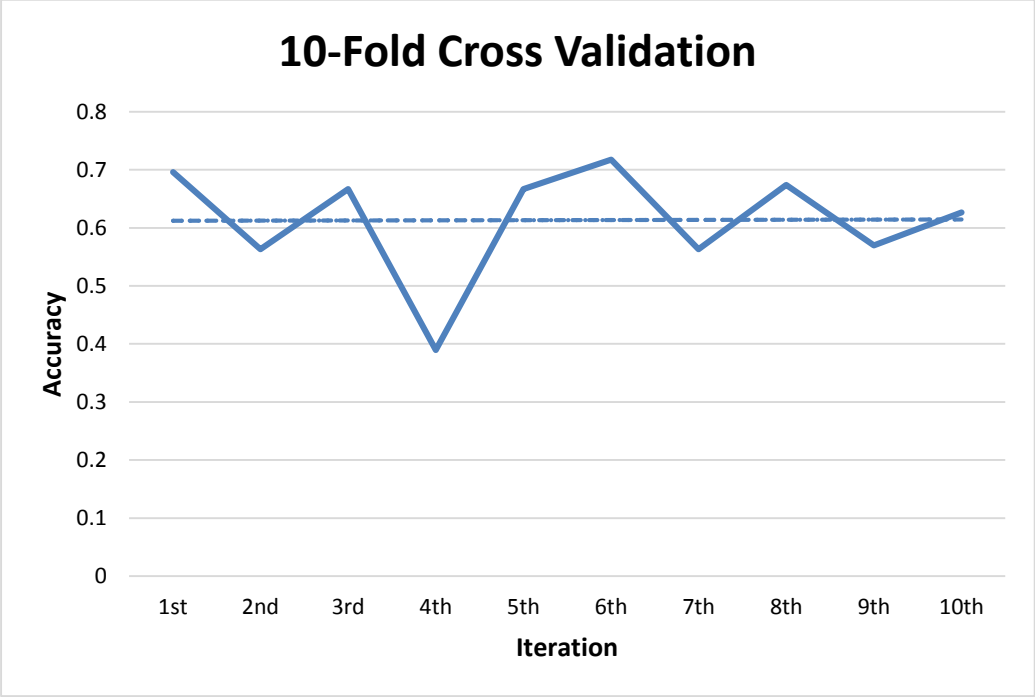


Figure 41 Tenfold CV graph with 200 sentences dataset

Table 7 below shows the results for each iteration of the 10-fold cross validation, as well as the aggregate accuracy of the 10-fold cross validation iterations with 200 sentences (i.e. sequences of dynamic gestures) dataset

Table 7 Results of the Tenfold CV with 200 sentences dataset

Iteration	**Accuracy**
1	0.696
2	0.563
3	0.667
4	0.389
5	0.667
6	0.717
7	0.563
8	0.674
9	0.570
10	0.626
Aggregate CV Accuracy	**0.613**

The tables above show our stochastic linear formal grammar is a quick learner. With only 100 command sequences it reaches a stable high performance, such that even when the dataset is doubled it shows minimal improvement.

It is important to note that without a stochastic linear formal grammar to capture the syntactic patterns of the data, the performance in the table above would drop to 0.08, which is equal to pure chance; since there are 12 dynamic gestures and the chance of predicting the correct one with no syntactic information is 1/12. Therefore the SLFG is performing 0.52 above the baseline, in other words 7.5 times higher in performance than the baseline.

5.3. Comparison with Other Approaches

We compare the results of our experiments with comparable previous works. Shiravandi et al. in [147], 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. in [148] 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 Fourier descriptor and is recognized by the neural network. The combined method of the motion trajectories and 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. in [149] did dynamic hand gesture recognition based on Speeded Up Robust Features (SURF) tracking. The main characteristic is that the dominant movement direction of matched SURF points in adjacent frames is used to help describing 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. in [150], proposed the hidden markov model (HMM) for hand gesture recognition. They consider 18 gestures for recognition. There recognition rate is 96.67%.

In [151], Pisharady et al. used dynamic time wrapping (DTW) and multi-class 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 8. As shown in the table, our experiment achieved the highest accuracy although we had the highest number of aggregate cases, parameters, and scenarios.

Compared to the previous works our work tackles the most complex task as shown by Table 8. As can be observed our task conditions vary among 24 different scenarios whereas the closest previous work tackles only 2. Despite the higher complexity of the task at arm our IP module alone reaches the performance of

97.22% which by itself outperforms the previous systems. In addition the complete DGLR system, by virtue of the use of the LRS module further improves the overall performance to 98.92% for humans and 97.42 for androids. (See Table 8).

Table 8 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[147]	12	24fps 320*240	No change	White	0	1	12	90
Motion Trajectories and Key Frames[148]	4	Not mention	No change	Black	0	1	4	96
SURF Tracking[149]	26	8-16fps 176*144	No change	Less clutter	1	2	52	84.6
Hidden Markov Model[150]	18	20fps 640*480	No change	Black	0	1	18	96.67
DTW and Multi class Probability[151]	12	Not mention	No change	Clutter	1	2	24	96.85
Our Approach for human (with grammar)	**12**	**30fps 160*120**	**Good light Bad light**	**Clutter White**	**9**	**24**	**288**	**98.92**
Our Approach for Android (with grammar)	**12**	**30fps 160*120**	**Good light Bad light**	**Clutter White**	**9**	**24**	**288**	**97.42**

5.4. Conclusions

The DGLR using formal grammars presented in this chapter uses a novel method to analyze the sentences (i.e., sequences of gestures) of the dynamic gesture language based on dynamic arm gestures. The LFG module checks if the commands are valid by devising a linear formal grammar. The reason for selecting a linear grammar as our syntactic pattern classifier, as opposed to more complex grammars, was stated. SLFG can operate on top of any pattern classification system, by taking the classification results as input and using a stochastic linear formal grammar to perform classification, where the previous system failed to do so. Therefore the SLFG module can be mounted on any dynamic arm gesture recognition system for any inherently sequential task including any type sign language recognition.

The SLFG module seals 61% prediction rate of non-recognized gestures for both human and android subjects. This prediction rate increase overall recognition rate of gestures performed by human as subject to 98.92% and for humanoid robot to 97.42%.

Chapter 6. Conclusions and Future Research

6.1. Conclusions

Dynamic arm gesturing represents a versatile non-verbal intuitive communication modality between humans as well as between humans and androids. As it can provide a worthy addition to the already accepted static hand signs (e.g. ASL JSL) it has recently started to attract the attention of researchers looking to expand the range of the human-centric communication modalities in a variety of emergent fields of activity where humans and intelligent androids are working together, such as manufacturing, industrial maintenance operations, disaster-management interventions, healthcare, eldercare, smart homes, etc.

The thesis presents a novel Dynamic Gesture Language Recognition (DGLR) system that we developed to support a Dynamic Gesture Language (DGL) prototype that we propose as an intuitive way of non-verbal and non-contact communication between humans and androids.

A consistent testing procedure was developed to measure the gesture recognition accuracy for each of the DGL gestures made by a human or by an android under different environmental conditions. In order to do this we developed an experimental setup consisting of a human-sized android able to execute in real time all the DGL dynamic arm gestures in similar way as humans do.

DGLR has a modular architecture consisting of an Image Processing (IP) module and a Linguistic Recognition System (LRS) module. This modularity allows to independently upgrade the modules so changes in one doesn't affect the other.

The IP module, which is based on a multi-scale local part model and on a bag-of-features and support vector machine classification, works well under a variety of lighting conditions and different transformations, occlusions and cluttered backgrounds. Experimental results have shown that the IP module has a

97.22% recognition rate in the case of human subjects and 93.40% recognition rate for android subjects.

The linguistic recognition system (LRS) module uses a novel formal grammar approach to accept DGL-wise valid sequences of gestures and reject invalid ones. LRS consists of two subsystems: one uses a Linear Formal Grammar (LFG) to derive the valid sequence of arm gestures and the other is using a Stochastic Linear Formal Grammar (SLFG) to predict occasional unrecognizable gestures. Experimental results have shown that the DGLR system had a slightly better overall performance in the case of human subjects (98.92% recognition rate) than for androids (97.42% recognition rate).

6.2. Future Work

The research will be expanded by a team of student who are developing a new experimental set up using a second-generation life-size android *InMoov* [133].

Further research will also be needed in order to integrate the speech recognition and the dynamic gesture recognition so they complement each other in a single complex recognition task. For instance if some command is unrecognizable to the android in one modality it can be disambiguated using the other modality. It will also provide for a more complex human like way of communication between robots and humans.

References

[1] E. M. Petriu, "Bio-Inspired Solutions for Intelligent Android Perception and Control," invited talk, "*ISTAS'13 IEEE Int. Symposium on Technology and Society," Toronto, ON, 2013, [Online]*. Available: http://www.site.uottawa.ca/~petriu/istas13-BioInAndroidPercepControl-130627a.pdf [Accessed 23.02.2015].

[2] E. M. Petriu, T.E. Whalen, "Computer-Controlled Human Operators," *IEEE Instrum. Meas. Mag.*, Vol. 5, No. 1, pp. 35 -38, 2002.

[3] Q. Chen, M.D. Cordea, E.M. Petriu, A.R. Varkonyi-Koczy, T.E. Whalen, "Human–Computer Interaction for Smart Environment Applications Using Hand Gestures and Facial Expressions," *Int. Journal Advanced Media and Communication*, Vol. 3, Nos. 1/2, pp. 95-109, 2009.

[4] 5 Ways Robots are Delivering Health Care in Saskatchewan *[Online]*. Available: http://www.cbc.ca/news/canada/saskatchewan/5-ways-robots-are-delivering-health-care-in-saskatchewan-1.2966190 [Accessed 23.02.2015].

[5] V. Unhelkar, J.A. Shah, "Challenges in Developing a Collaborative Robotic Assistant for Automotive Assembly Lines," *Proc. Tenth Annual ACM/IEEE Int. Conf. Human-Robot Interaction - HRI'15*, pp. 239-240, Portland, Oregon, USA, March 2015.

[6] P.A. Lasota, J.A. Shah, "Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human-Robot Collaboration," *Human Factors*, Vol. 57, No. 1, pp. 21-33, Feb 2015.

[7] S. Nikolaidis, K. Gu, R, Ramakrishnan, J.A. Shah, "Efficient Model Learning from Joint-Action Demonstrations for Human-Robot Collaborative Tasks," *Proc. Tenth Annual ACM/IEEE Int. Conf. Human-Robot Interaction - HRI'15*, pp. 189-196, Portland, Oregon, USA, March 2015.

[8] F. Shi, E. M. Petriu, A. Cordeiro, and N. D. Georganas, "Human Action Recognition from Local Part Model," *Haptic Audio Visual Environments and Games(HAVE),* pp. 35-38, 2011.

[9] M. R. Abid, E.M. Petriu, E. Amjadian, "Dynamic Sign Language Recognition for Smart Home Interactive Application Using Stochastic Linear Formal Grammar," *IEEE Trans. Instrum. Meas*., vol.64, no.3, pp.596,605, March 2015.

[10] M. R. Abid, P.E. Meszaros, R.F. da Silva, E.M. Petriu, "Dynamic Hand Gesture Recognition for Human-Robot and Inter-Robot Communication," *Proc. CIVEMSA 2014 - IEEE Int. Conf. on Computational Intelligence and Virtual Environments for Meas. Systems and Applications*, pp. 12-17, Ottawa, ON, Canada, May 2014

[11] M. R. Abid, L.B. Santiago Melo, E.M. Petriu, “Dynamic Sign Language and Voice Recognition for Smart Home Interactive Application,” *Proc. MeMeA2013, 8th IEEE Int. Symp. on Medical Measurement and Applications*, pp. 139-144, Ottawa, ON, Canada, May 2013.

[12] M. R. Abid, F. Shi, E.M. Petriu, “Dynamic Hand Gesture Recognition from Bag-of-Features and Local Part Model,” *Proc. HAVE 2012 - IEEE Int. Symp. Haptic Audio Visual Environments and Games, 78 – 82*, Munich, Germany, Oct. 2012.

[13] Q. Chen, F. Malric, Y. Zhang, M. Abid, A. Cordeiro, E.M. Petriu, N.D. Georganas, "Interacting with Digital Signage Using Hand Gestures", *Proc. ICIAR 2009, Int. Conf. Image Analysis and Recognition, (M. Kamel and A. Campilho - Eds), Lecture Notes in Computer Science Vol. LNCS 5627,* pp. 347-358, Springer, Berlin/Heidelberg, 2009.

[14] A. Corradini "Real-time Gesture Recognition by Means of Hybrid Recognizers," Lecture Notes in Computer Science, Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction, vol. 2298, pp.34 -46 2001.

[15] R. H. Liang and M.Ouhyoung, "A Real Time Continuous Gesture Recognition System for Sign Language," *in Proc . 3rd International Conference on Automatic Face and Gesture Recognition,* 1998, pp. 558-565.

[16] V. I. Pavlovic, R. Sharma, T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, july 1997 pp. 677-695.

[17] M. Alsheakhali, A. Skaik, M. Aldahdouh, M. Alhelou, "Hand Gesture Recognition System," *Computer Engineering Department, The Islamic University of Gaza Strip,* Palestine, 2011.

[18] S.Sidney Fels and G. E. Hinton, "Glove-Talk: A Neural Network Interface Between a Data-Glove and a Speech Synthesizer," *IEEE Transactions on Neural Networks,* Vol. 3, No. 6, November 1992 pp.1-7.

[19] M. Ali Qureshi, A. Aziz, M. Ammar Saeed, M. Hayat, "Implementation of an Efficient Algorithm for Human Hand Gesture Identification," *Dept. Electronic Engineering, University College of Engineering & Technology, The Islamia University of Bahawalpur, Bahawalpur Pakistan.*

[20] D.J. Sturman and D. Zeltzer, "A Survey of Glove-Based Input, "*IEEE Computer Graphics and Applications,* vol. 14, pp. 30-39, Jan. 1994.

[21] Kumo, Yoshinori, T. Ishiyama, KH. Jo, N. Shimada and Y. Shirai, "Vision-Based Human Interface System: Selectively Recognizing Intentional Hand Gestures," *In Proceedings of the IASTED International Conference on Computer Graphics and Imaging,* 219-223, 1998.

[22] Utsumi, Akira, J. Kurumisawa, T. Otsuka, and J. Ohya, "Direct Manipulation Scene Creation in 3D*," SIGGRAPH'97 Electronic Garden,* 1997.

[23] Davis, James, and M. Shah, "Gesture Recognition. Technical Report," *Department of Computer Science, University of Central Florida,* CS-TR-93-11, 1993.

[24] Starner, Thad, "Visual Recognition of American Sign Language Using Hidden Markov Models," *Master's thesis, Massachusetts Institute of Technology,* 1995.

[25] Dorner, Brigitte, "Chasing the Colour Glove: Visual Hand Tracking," *Master's thesis, Simon Fraser University,* 1994.

[26] M. de La Gorce, N. Paragios, D. J. Fleet, "Model based Hand Tracking with Texture, Shading and Self Occlusions," *In IEEE Conference on Computer Vision and Pattern Recognition,* 2008, pp. 1-8.

[27] B. Rosenhahn, T. Brox, J. Weickert, "Three-Dimensional Shape Knowledge for Joint Image Segmentation and Pose Tracking," *International Journal of Computer Vision*, 73 (2007) 243–262.

[28] J. Rehg and T. Kanade. Digiteyes, "Vision-based Hand Tracking for Human-Computer Interaction," *In Workshop on Motion of Non-Rigid and Articulated Bodies,* pages 16-24, Austin Texas, November 1994.

[29] Y. Wu, J. Lin, and T. Huang, "Capturing Natural Hand Articulation". *In IEEE International Conference on Computer Vision II*, 426–432, 2001.

[30] B. Stenger, P. Mendonca, & R. Cipolla, "Model-Based 3D Tracking of an Articulated Hand," *In IEEE Conference on Computer Vision and Pattern Recognition,* 310–315, 2001.

[31] H. Zhou, D. J. Lin, and T. S. Huang, "Static Hand Gesture Recognition Based on Local Orientation Histogram Feature Distribution Model," *In Proc. Conference on Computer Vision and Pattern Recognition Workshop*, vol. 10, 2004, pp. 161–169.

[32] F. Chen, C.M. Fu, and C.L. Huang, "Hand Gesture Recognition Using a Real-Time Tracking Method and Hidden Markov Models," *Image and Vision Computing,* 21(8):745-758, 2003.

[33] C. Travieso, J. B. Alonso, and M. A. Ferrer, "Sign Language to Text by SVM," *In Proc. Seventh International Symposium on Signal Processing and Its Applications,* volume 2, pages 435-438 vol.2, 2003.

[34] B. Stenger, "Template Based Hand Pose Recognition Using Multiple Cues", *In Proc. 7th Asian Conference on Computer Vision: ACCV* 2006.

[35] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand Gesture Recognition Using Multi Scale Colour Features, Hierarchical Models and Particle Filtering," *In Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on,* pages 405-410, 2002.

[36] C. W. Ng and S. Ranganath, "Gesture Recognition via Pose Classification," *In Proc. 15th International Conference on Pattern Recognition,* vol. 3, 2000, pp. 699-704.

[37] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady, "A Linguistic Feature Vector for the Visual Interpretation of Sign Language," *In Proc. European Conference on Computer Vision,* 2004, pp. 391–401.

[38] K. Oka, Y. Sato and H. Koike, "Real-time Fingertip Tracking and Gesture Recognition," *In Proc. IEEE Computer Graphics and Applications,* vol. 22, no. 6, 2002, pp. 64–71.

[39] Z. Zhang, Y. Wu, Y. Shan, and S. Shafer, "Visual Panel: Virtual Mouse Keyboard and 3D Controller with an Ordinary Piece of Paper," *In Proc. Workshop on Perceptive User Interfaces*, 2001.

[40] S. Malik, C. McDonald, and G. Roth, "Hand Tracking for Interactive Pattern-based Augmented Reality," *In Proc. International Symposium on Mixed and Augmented Reality,* 2002.

[41] Q. Yuan, S. Sclaroff, and V. Athitsos, " Automatic 2D Hand Tracking in Video Sequences," *In Proc. IEEE Workshops on Application of Computer Vision, 2005,* pp. 250–256.

[42] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time Tracking of Non-Rigid Objects Using Mean Shift," *In Proc. IEEE Computer Vision and Pattern Recognition,* vol. 2, no. 2, 2000, pp. 142–149.

[43] G. Bradski, "Real Time Face and Object Tracking as a Component of a Perceptual User Interface," *In Proc. IEEE Workshop on Applications of Computer Vision,* 1998, pp. 214–219.

[44] D. G. Lowe, "Distinctive Image Features From Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60(2):91-110, 2004.

[45] P. Viola, M. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision* 57, 2 (May 2004), 137–154.

[46] M. Kolsch and M. Turk, "Robust Hand Detection," In Proc. 6th IEEE Conference on Automatic Face and Gesture Recognition, 2004.

[47] M. Kolsch, M. Turk, and T. Hollerer, "HandVu Vision-based Hand Gesture Interface," [Online]. Available: *http://ilab.cs.ucsb.edu/projects/mathias/handvu ilab.html*

[48] M. Kolsch, "Vision Based Hand Gesture Interfaces for Wearable Computing and Virtual Environments," *Ph.D. dissertation, University of California, Santa Barbara,* 2004. [Online]. Available: *http://www.movesinstitute.org/_kolsch/publications.html*

[49] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience,* 3(1), 1991.

[50] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," *Journal of the Optical Society of America,* 4:519-524, March 1987.

[51] H. Murase and S. Nayar, "Visual Learning and Recognition of 3d Objects from Appearance," *International Journal of Computer Vision*, 14:5-24, 1995.

[52] P. Morguet and M. Lang, "Spotting Dynamic Hand Gestures in Video Image Sequences using Hidden Markov Models," *IEEE Image Processing,* pp. 193-197, 1998.

[53] X. Wang, M. Xia, H. Cai, Y. Gao and C. Cattani, "Hidden Markov Models based Dynamic Hand Gesture Recognition," *Mathematical Problems in Engineering*, 2012.

[54] H. Suk, B. Sin and S. Lee, "Hand Gesture Recognition Based on Dynamic Bayesian Network Framework," *Pattern Recognition*, pp. 3059-3072, 2010.

[55] C. David,V. Gui, P. Nisula, and V. Korhonen, "Dynamic Hand Gesture Recognition for Human-Computer Interactions," *SACII*, 2011.

[56] A. Ramamoorthy, N. Vaswani, S. Chaudhury, and S. Banerjee, "Recognition of Dynamic Hand Gestures," *Pattern Recognition*, vol. 36, pp. 2069–2081, 2003.

[57] J. Anderson, "An Introduction to Neural Networks," *The MIT Press*, 1995.

[58] G. Hua and Y.Yes Wu, "Multi-scale Visual Tracking by Sequential Belief Propagation," In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.

[59] S. Kettebekov, M. Yeasin, and R. Sharma, "Prosody Based Audiovisual Coanalysis for Coverbal Gesture Recognition," *Multimedia, IEEE Transactions on*, 7(2):234-242, 2005.

[60] A. El-Sawah, N. Georganas, and E. Petriu, "A Prototype for 3-D Hand Tracking and Gesture Estimation," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 8, pp. 1627–1636, Aug. 2008.

[61] E. Ong and R. Bowden, "Detection and Segmentation of Hand Shapes Using Boosted Classifiers," *In Proc. IEEE 6th International Conference on Automatic Face and Gesture Recognition,* 2004, pp. 889–894.

[62] S. Gutta, H. Huang, F. Imam, and H. Wechsler, "Face and Hand Gesture Recognition using Hybrid Classifiers," *In Proc. Second International Conference on Automatic Face and Gesture Recognition,* pages 164-169, 1996.

[63] Z. Hang and R. Qiuqi, "Visual Gesture Recognition with Color Segmentation and Support Vector Machines," *In Proc. 7th International Conference on Signal Processing ICSP '04*, volume 2, pages 1443-1446 vol.2, 2004.

[64] H. Habib and M. Mufti, "Real Time Mono Vision Gesture based Virtual Keyboard System," *Consumer Electronics, IEEE Transactions on*, 52(4):1261-1266, 2006.

[65] V. Rao and C. Mahanta, "Gesture Based Robot Control," In Proc. Fourth International Conference on Intelligent Sensing and Information Processing ICISIP 2006, pages 145-148, 2006.

[66] M. Su, "A Fuzzy Rule-based Approach to Spatio-temporal Hand Gesture Recognition," *IEEE Transactions on Systems, Man, and Cybernetics Part C*, 30(2):276-281, 2000.

[67] Russell, Stuart, and Peter Norvig, "Artificial Intelligence: A Modern Approach," *Prentice Hall, Englewood Cliffs, NJ*, 1995.

[68] Krose, Ben J. A., and P. Patrick van der Smagt, "An Introduction to Neural Networks," *University of Amsterdam*, 1995.

[69] 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.

[70] S. D. Gawande and N. R. Chopde, "Neural Network Based hand Gesture Recognition," *International journal of Emerging Research in Management & Technology,* ISSN: 2278-9359(Volume-2, Issue-3), March 2013.

[71] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-Based Learning Algorithms." *Machine Learning,* 6(1):37–66.

[72] S. Ali, M. Shah, "Human Action Recognition in Videos using Kinematic Features and Multiple Instance Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2010.

[73] V. Vapnik. "Statistical Learning Theory," *Wiley-Interscience*, 1998.

[74] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Fifth Annual Workshop of Computational Learning Theory*, 5, 144-152, Pittsburgh, ACM.

[75] J. Weston, C. Watkins, "Multi-class Support Vector Machines," *Proceedings of ESANN99, M. Verleysen, Ed., Brussels, Belgium*, 1999.

[76] 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.

[77] C. W. Hsu and C. J. Lin, "A Comparison of Methods for Multi-class Support Vector Machines," *IEEE 373 Transactions on Neural Networks,* 13(2002), 415-425.

[78] J. H. Friedman, "Another Approach to Polychotomous Classification," *Technical Report. Department of Statistics, Stanford University*, 1997.

[79] R. Cutler, M. Turk, "View based Interpretation of Real Time Optical Flow for Gesture Recognition," *3rd IEEE Conf. on Face and Gesture Recognition, Nara, Japan,* April 1998.

[80] R. Cutler, and M. Turk, "View-Based Interpretation of RealTime Optical Flow for Gesture Recognition," *In IEEE Int. Conf. on Automatic Face and Gesture Recognition*, (1998) 416–421.

[81] S. Mu-Chun, "A Fuzzy Rule-Based Approach to SpatioTemporal Hand Gesture Recognition," *IEEE Trans. on Systems, Man and Cybernetics, Appl. and Review*, 30 (2000).

[82] S. M. Hassan, A. Farouk Al-Sadek, E. E. Hemayed, " Rule-based Approach for Enhancing the Motion Trajectories in Human Activity Recognition," *Intelligent Systems Design and Applications (ISDA)*, 10 th International Conference, pp. 829-834, Cairo, 2010.

[83] J.Y. Chang, J.J. Shyu, C.W. Cho. "Fuzzy Rule Inference Based Human Activity Recognition." *IEEE International Symposium on Intelligent Control,* 2009.

[84] S. Zou, H. Xiao, H. Wan, X. Zhou, "Vision based Hand Interaction and Its Application in Pervasive Games," *In: Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, Yokohama, Japan, pp. 157–162 (2009).

[85] K. S. Fu, "Syntactic Pattern Recognition and Applications," *New Jersey: Prentice Hall*, 1982.

[86] M. Sonka, V. Hlavac, and R. Boyle, "Image Processing, Analysis, and Machine Vision," *PWS Publishing,* 1999.

[87] N. Chomsky, "Syntactic Structures," *The Hague: Mouton*, 1966.

[88] D. Grune and C. J. H. Jacobs, "Parsing Techniques: A Practical Guide," *Ellis Horwood,* 1991.

[89] C. Hand, I. Sexton, and M. Mullan, "A Linguistic Approach to the Recognition of Hand Gestures," *In IEE Ergonomics Society Proc. Designing Future Interaction Conference,* 1994.

[90] K. G. Derpanis, R. P. Wildes, and J. K. Tsotsos, "Hand Gesture Recognition within a Linguistics-Based Framework," *In Proc. European Conference on Computer Vision,* 2004, pp. 282–296.

[91] M. S. Ryoo and J. K. Aggarwal, 'Recognition of Composite Human Activities through Context-Free Grammar based Representation," *In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 2, 2006, pp. 1709–1718.

[92] I. Laptev and T. Lindeberg, "Space-Time Interest Points," *In Computer Vision, 2003. ICCV 2003. IIEEE 9th International Conference on*, pages 432 - 439, 2003.

[93] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition Via Sparse Spatio-Temporal Features," *In Proc. 2nd Joint IEEE Int Visual Surveillance and Performance Evaluation of Tracking and Surveillance Workshop*, pages 65 - 72, 2005.

[94] H. Wang, M. Muneeb Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of Local Spatio-Temporal Features for Action Recognition," *In BMVC,* pages 127 - 137, 2009.

[95] H. Wang, A. Klaser, C. Schmid, and Cheng-Lin Liu, "Action Recognition by Dense Trajectories," *In CVPR*, pages 3169-3176, 2011.

[96] R. Messing, C. Pal, and H. Kautz, "Activity Recognition Using the Velocity Histories of Tracked Keypoints," *In Computer Vision, 2009 IEEE 12th International Conference on*, pages 104-111, 2009.

[97] P. Scovanner, S. Ali, and M. Shah, "A 3-Dimensional SIFT Descriptor and its Application to Action Recognition," *In Proceedings of the 15th International Conference on Multimedia*, pages 357-360. ACM, 2007.

[98] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: Action Recognition Through the Motion Analysis of Tracked Features," *In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 514-521, 2009.

[99] P. Matikainen, M. Hebert, and R. Sukthankar, "Representing Pairwise Spatial and Temporal Relations for Action Recognition," *In Computer Vision ECCV 2010,* pages 508-521. Springer, 2010.

[100] I. Laptev and T. Lindeberg, "Local Descriptors for Spatio-Temporal Recognition," *In Spatial Coherence for Visual Motion Analysis*, pages 91-103, Springer, 2006.

[101] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, " Learning Realistic Human Actions from Movies," *In CVPR*, pages 1-8, 2008.

[102] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *In CVPR,* volume 1, pages 886-893. IEEE, 2005.

[103] A. Klaser, M. Marszalek, and C. Schmid, "A Spatio-Temporal Descriptor Based on 3D-Gradients," *In BMVC*, pages 995-1004, 2008.

[104] G. Salton, "Automatic Information Organization and Retrieval," *McGraw Hill Text*, 1968.

[105] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Proc. 10th European Conf. on Machine Learning, Dorint-Parkhotel, Germany,* April 1998, pp. 137–142.

[106] S. Tong, D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *J. Mach. Learn. Res.*, 2002, 2, pp. 45–66.

[107] H. Lodhi, J. Shawe-Taylor, N. Christianini, C. Watkins, "Text Classification Using String Kernels," *J. Mach. Learn. Res.*, 2002, 2, pp. 419–444.

[108] N. Cristianini, J. Shawe-Taylor, H. Lodhi, "Latent Semantic Kernels," *J. Intell. Inf. Syst.*, 2002, 18, (2), pp. 127–152.

[109] M. Jain, H._e J_egou, and P. Bouthemy, "Better Exploiting Motion for Better Action Recognition," *In CVPR*, 2013.

[110] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell and G. Coleman, "Detection and Explanation of Anomalous Activities: Representing Activities as Bags of Event n-grams," *In Computer Vision and Pattern Recognition, 2005. IEEE Computer Society Conference on*, volume 1, pages 1031-1038, 2005.

[111] S. Maybank and O. D. Faugeras. "A Theory of Self Calibration of a Moving Camera," *Int. Journal of Computer Vision*, 8(2), pp.123–151, August 1992.

[112] O. D. Faugeras, Q. T. Luong, and S. J. Maybank, “Camera Self-Calibration: Theory and Experiments,” *In G. Sandini, editor, Proc. 2nd European Conf. On Comp Vision, Lecture Notes in Comp. Science 588*, pp. 321–334. Springer–Verlag, May 1992.

[113] B. M. Mooring, Z. S. Roth, and M. R. Driels, "Fundamentals of Manipulator Calibration," *New York: Wiley Interscience*, 1991.

[114] H. W. Stone, "Statistical Performance Evaluation of the S-model Arm Signature Identification Technique," *Proc. IEEE International Conference on Robotics and Automation*, 1988, pp. 939-946.

[115] H. Q. Zhuang, Z. S. Roth, "A Linear Solution to the Kinematic Parameter Identification of Robot Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9(2), 1993, pp. 174-185.

[116] M. Rüther, M. Lenz and H. Bischof, "The Narcissistic Robot: Robot Calibration Using a Mirror," *11th Int. Conf. Control, Automation, Robotics and Vision, Singapore,* 7-10th December 2010, pp. 169-174.

[117] H. X. Wang, S. H. Shen and X. Lu, "A Screw Axis Identification Method for Serial Robot Calibration Based on the POE Model," *Industrial Robot,* vol. 39(2), 2012, pp. 146 – 153.

[118] http://www.juniperresearch.com, Retrieved March 03, 2015.

[119] http://www.biometricupdate.com/201404/explainer-gesture-recognition, Retrieved March 03, 2015.

[120] T. Bramwell, “E3: MS execs: Natal not derived from 3DV,” *Eurogamer. Eurogamer Network*. Retrieved March 03, 2015.

[121] J. Lee, "Project Natal,” *Procrastineering*. Retrieved March 03, 2015.

[122] D. Takahashi, “Microsoft Games Exec Details How Project Natal was Born,”.*VentureBeat*. Retrieved March 03, 2015.

[123] V. Frati, D. Prattichizzo, “Using Kinect for Hand Tracking and Rendering in Wearable Haptics,” *IEEE World Haptics Conference*, 2011.

[124] M. Bergh, D. Carton, R. Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Gool, M. Buss, “Real-time 3D Hand Gesture Interaction with a Robot for Understanding Directions from Humans,” *Proceedings of 20th IEEE International Symposium on Robot and Human Interactive Communication*, Atlanta Georgia , 2011.

[125] J. Raheja, A. Chaudhary, K. Singal, “Tracking of Fingertips and Centres of Palm using KINECT,” *IEEE Third International Conference on Computational Intelligence*, Modelling & Simulation, 2011.

[126] Z. Ren, J. Yuan, Z. Zhang, “Robust Hand Gesture Recognition Based on Finger-Earth Mover’s Distance with a Commodity Depth Camera,” *Proc. of ACM Intl. Conf. on Multimedia (ACM MM 11), Scottsdale, Arizona, USA*, Nov. 28-Dec. 1, 2011.

[127] B. Dudley, "E3: New info on Microsoft's Natal - How It Works, Multiplayer and PC Versions," *Brier Dudley's Blog*. The Seattle Times. Retrieved March 03, 2015.

[128] G. Simei Wysoski, V. Marcus Lamar, K. Susumu, I. Akira, "A Rotation Invariant Approach On Static-Gesture Recognition Using Boundary Histograms And Neural Networks," *IEEE Proceedings of the 9th International Conference on Neural Information Processing, Singapura*. 2002.

[129] M. Kouichi, T. Hitomi, "Gesture Recognition using Recurrent Neural Networks," *ACM Conference on Human Factors in Computing Systems: Reaching Through Technology (CHI '91)*, pp. 237-242, 1991.

[130] A. Malima, E. Ozgur and M. Çetin, "A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control," *IEEE 14th Conference on Signal Processing and Communications Applications,* pp. 1- 4, 2006.

[131] S. E. Ghobadi, O. E. Loepprich, F. Ahmadov, J. Bernshausen, K. Hartmann and O. Loeld, "Real Time Hand Based Robot Control Using Multimodal Images," *International Journal of Computer Science IJCS*. Vol 35(4), 2008.

[132] Dr. Robot, [*Online*]. Available: http://www.drrobot.com/products_hawk.asp

[133] InMoov. (2013, 12 27). *InMoov Robot Hand* [Online]. Available: http://www.thingiverse.com/thing:17773.

[134] G. Berry, "A Multimodal Human Computer Intelligent Interaction Test Bed with Applications," *Dept. of ECE, University of Illinois at Urbana-Champaign, MS thesis*, 1998.

[135] Lucente, Mark, Z. Gert-Jan and G. D. Andrew, "Visualization Space: A Testbed for Deviceless Multimodal User Interface," *In Intelligent Environments 98, AAAI Spring Symposium Series*, 87-92, 1998.

[136] Y. Kuno, T. Murashima,, N. Shimada, Y. Shirai, "Intelligent Wheelchair Remotely Controlled by Interactive Gestures," *In Proceedings of 15th International Conference on Pattern Recognition (Barcelona, Sept. 3–7, 2000)*, 672–675.

[137] C. Graetzel, T. Fong, C. Grange, C. Baur, "A Non-Contact Mouse for Surgeon-Computer Interaction," *Technology and Health Care 12*, 3 (Aug. 24, 2004), 245–257.

[138] J. Wachs, M. Kölsch, H. Stern, Y. Edan, "Vision-Based Hand-Gesture Applications," *Communications of the ACM*, Volume 54 Issue 2, February 2011.

[139] K. George, "BodyMaps", 24 april 2013, available: [*online*] "http://www.healthline.com/human-body-maps/arm".

[140] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *In Proc. IJCV,* Nov. 2004, pp. 91–110.

[141] 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.

[142] V. Vapnik, "The Nature of Statistical Learning Theory," *New York:Springer,* 1995.

[143] MakerBot® Industries, LLChttp://store.makerbot.com/replicator2x [online].

[144] Custom Entertainment Solutions, Inc. . (2013, 12 27). *MechaTE Robot Hand Limited Edition* [Online]. Available: http://www.animatronicrobotics.com/shopping/components/mechate-robot-hand-limited-edition/1-3.html.

[145] W. J.M. Levelt, "An Introduction to the Theory of Formal Languages and Automata," John Benjamins B.V. 2008.

[146] N. Comsky, "On Certain Formal Properties of Grammars," *Information And Control 2*, 137-167 (1959)

[147] S. Shiravandi, M. Rahmati, and F. Mahmoudi. "Hand Gestures Recognition Using Dynamic Bayesian Networks," *In AI & Robotics and 5th RoboCup Iran Open International Symposium (RIOS),* 2013 3rd Joint Conference. Apr 2013, pp. 1-6.

[148] 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.

[149] 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.

[150] 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.

[151] 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.

Appendix A: UML Sequence Diagrams for Dynamic Gestures

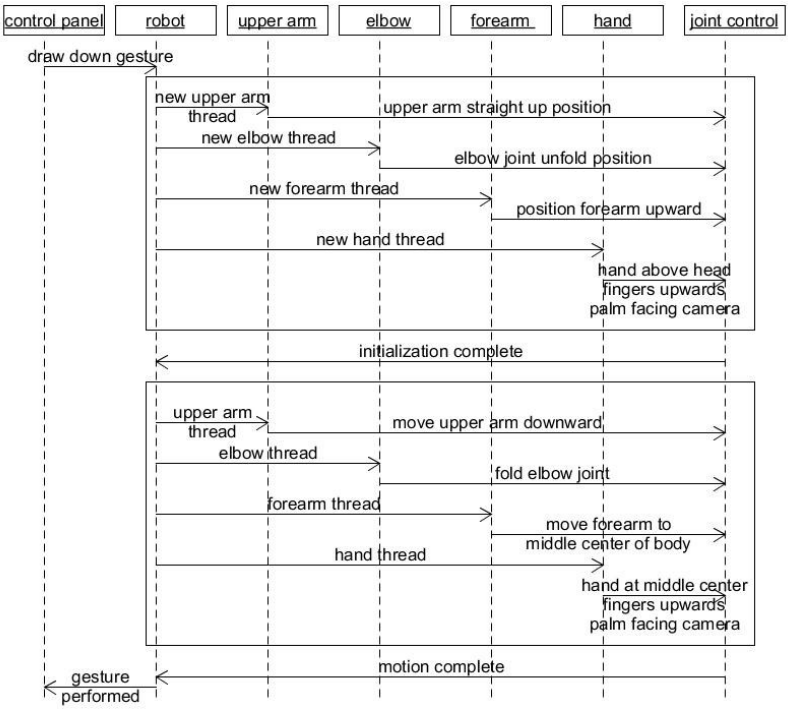


Figure 42 UML sequence diagram for "down" gesture

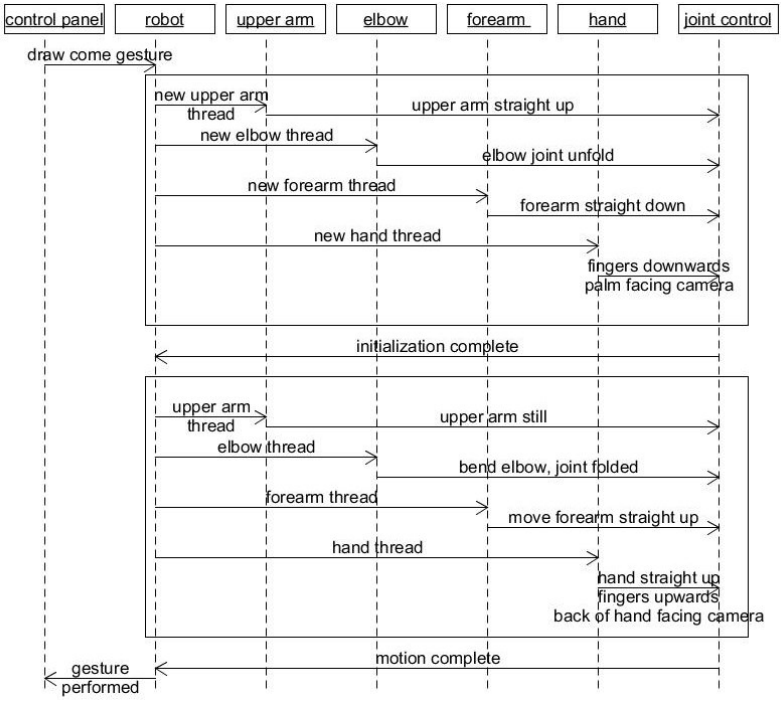


Figure 43 UML sequence diagram for “come” gesture

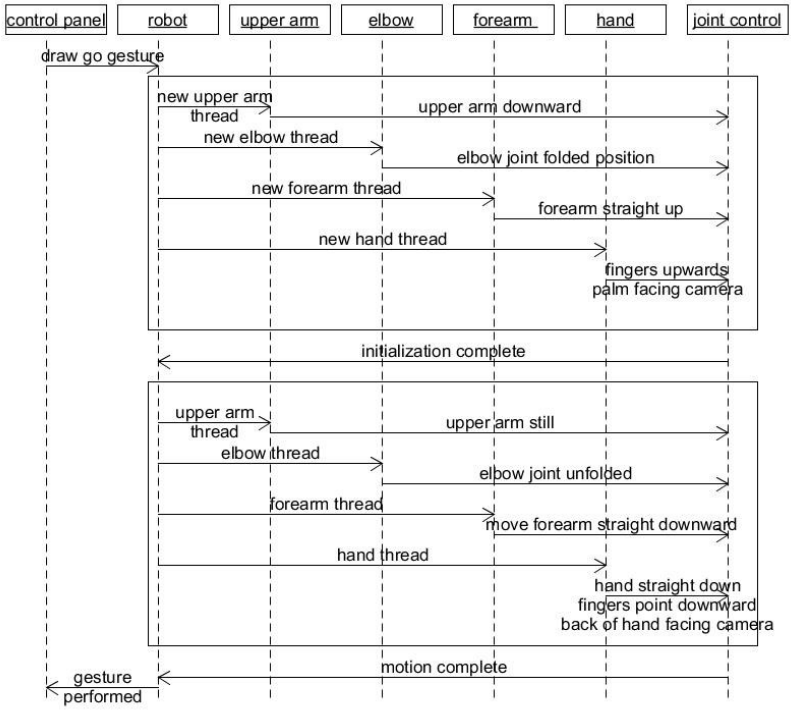


Figure 44 UML sequence diagram for “go” gesture

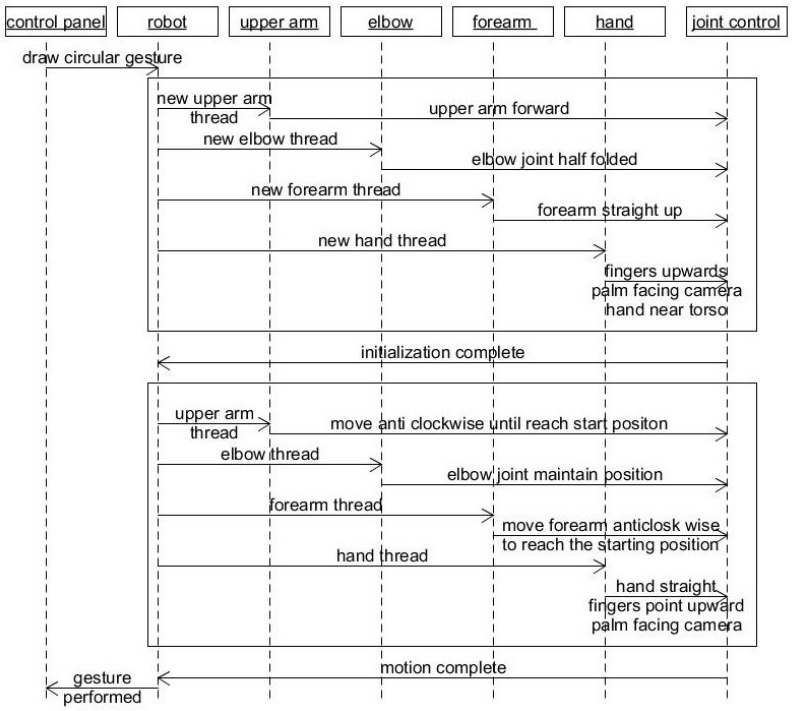


Figure 45 UML sequence diagram for “circular” gesture

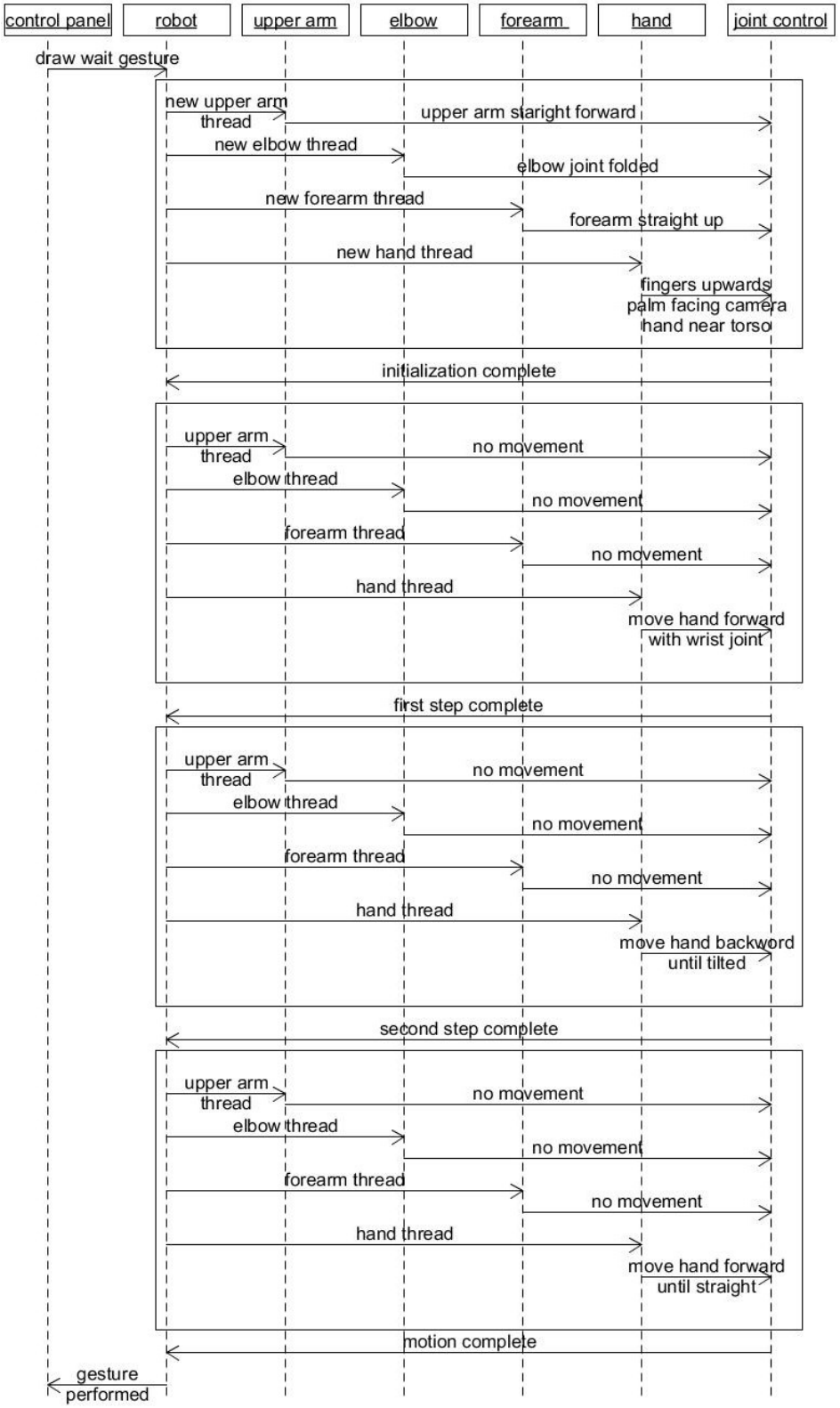


Figure 46 UML sequence diagram for “wait” gesture

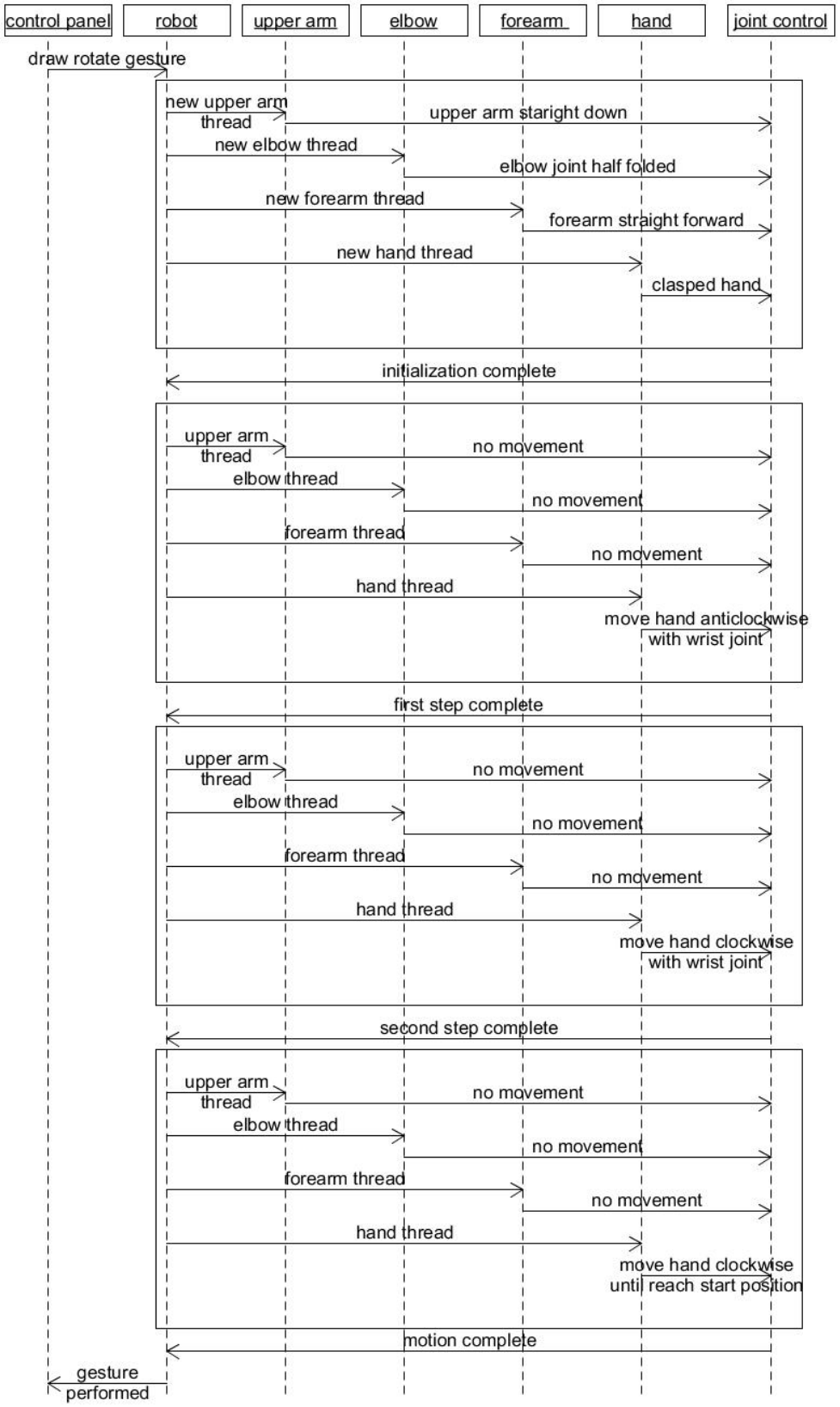


Figure 47 UML sequence diagram for “rotate” gesture

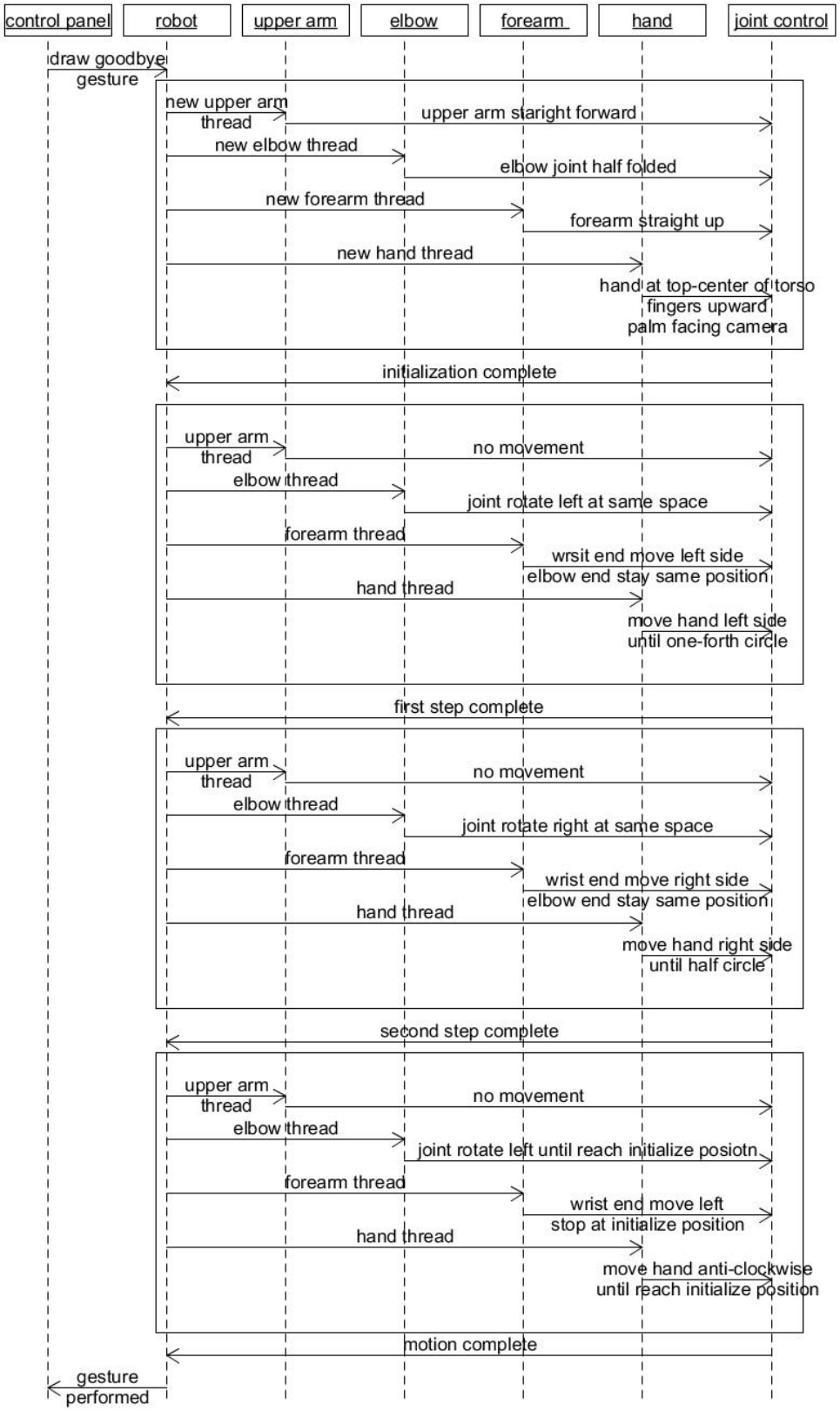


Figure 48 UML sequence diagram for “bye” gesture

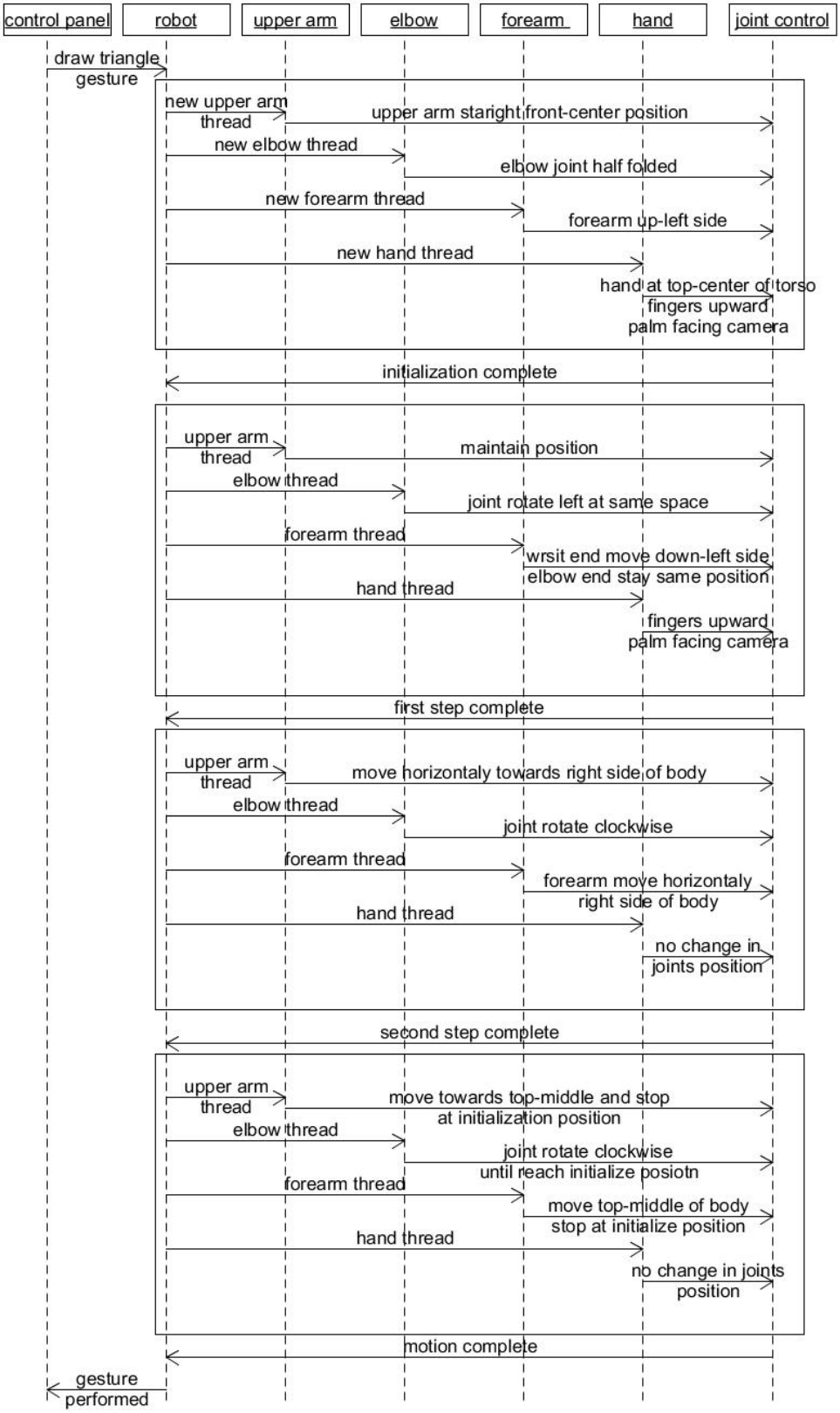


Figure 49 UML sequence diagram for “triangle” gesture

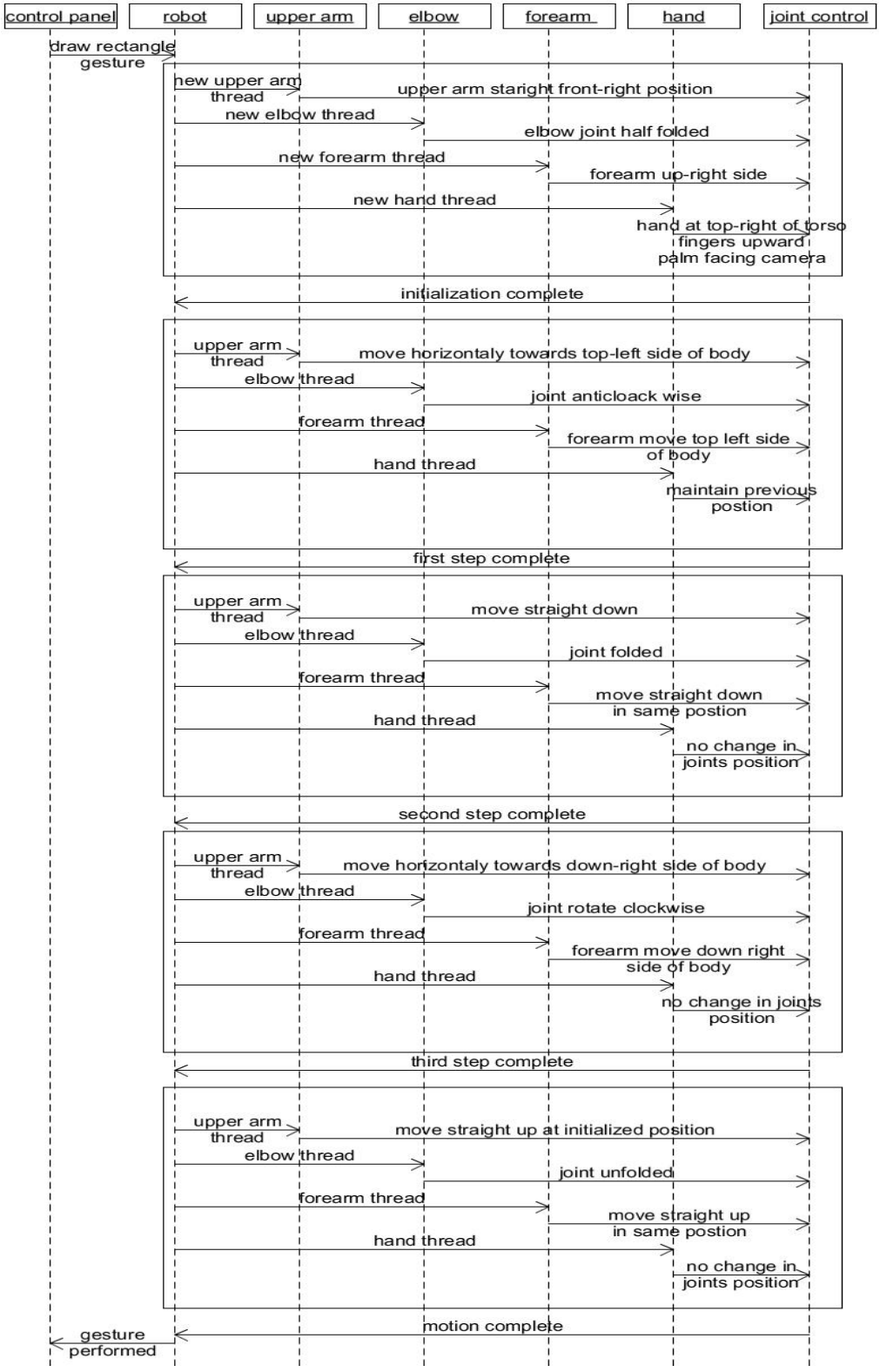


Figure 50 UML sequence diagram for "rectangle" gesture

Appendix B: Training Dataset for SLFG Module

This is the corpus of the language created by grammar interface. We used this dataset to train our SLFG module.

```
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up look_up move_right_arm_toward_left
move_right_arm_toward_right open_both_hands close_both_hands
bye
Wake_up raise_both_arms bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up move_right_arm_toward_left look_up raise_both_arms
open_both_hands bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_right_arm lower_both_arms bye
Wake_up raise_both_arms lower_both_arms raise_right_arm
raise_both_arms bye
```

```
Wake_up raise_both_arms open_both_hands lower_right_arm
lower_both_arms close_both_hands bye
Wake_up raise_both_arms open_both_hands lower_right_arm
lower_both_arms close_both_hands bye
Wake_up look_up look_down move_right_arm_toward_left
raise_right_arm lower_both_arms bye
Wake_up look_up raise_right_arm open_both_hands look_down
move_right_arm_toward_left lower_right_arm bye
Wake_up open_both_hands raise_right_arm
move_right_arm_toward_left look_up close_both_hands
lower_both_arms bye
Wake_up look_up look_down bye
Wake_up raise_right_arm move_right_arm_toward_left look_up
open_both_hands bye
Wake_up look_up move_right_arm_toward_left
move_right_arm_toward_right raise_right_arm lower_right_arm
bye
Wake_up raise_both_arms lower_right_arm lower_both_arms
raise_right_arm bye
Wake_up open_both_hands raise_right_arm close_both_hands
lower_right_arm bye
Wake_up look_up raise_right_arm look_down lower_right_arm bye
Wake_up move_right_arm_toward_left raise_right_arm
move_right_arm_toward_right look_up bye
Wake_up open_both_hands raise_both_arms close_both_hands
lower_both_arms bye
Wake_up open_both_hands raise_right_arm lower_both_arms
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up move_right_arm_toward_left look_up raise_both_arms
open_both_hands bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
```

```
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up look_up move_right_arm_toward_left
move_right_arm_toward_right open_both_hands close_both_hands
bye
Wake_up raise_both_arms bye
Wake_up raise_both_arms open_both_hands look_up
move_right_arm_toward_left move_right_arm_toward_right
look_down lower_both_arms raise_right_arm raise_both_arms
lower_both_arms raise_right_arm raise_both_arms
lower_right_arm close_both_hands raise_both_arms look_up
open_both_hands lower_both_arms raise_right_arm
lower_both_arms raise_both_arms lower_right_arm
close_both_hands look_down open_both_hands close_both_hands
bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up raise_right_arm lower_right_arm look_up
open_both_hands move_right_arm_toward_left
move_right_arm_toward_right raise_both_arms lower_both_arms
raise_both_arms lower_right_arm look_down
move_right_arm_toward_left close_both_hands lower_both_arms
move_right_arm_toward_right look_up look_down raise_both_arms
lower_right_arm open_both_hands move_right_arm_toward_left
move_right_arm_toward_right look_up close_both_hands
raise_right_arm look_down move_right_arm_toward_left look_up
open_both_hands lower_both_arms bye
Wake_up move_right_arm_toward_left look_up raise_both_arms
open_both_hands bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_right_arm lower_both_arms bye
Wake_up raise_both_arms lower_both_arms raise_right_arm
raise_both_arms bye
Wake_up raise_both_arms open_both_hands lower_right_arm
lower_both_arms close_both_hands bye
Wake_up raise_both_arms open_both_hands lower_right_arm
lower_both_arms close_both_hands bye
Wake_up look_up look_down move_right_arm_toward_left
raise_right_arm lower_both_arms bye
Wake_up look_up raise_right_arm open_both_hands look_down
move_right_arm_toward_left lower_right_arm bye
Wake_up open_both_hands raise_right_arm
move_right_arm_toward_left look_up close_both_hands
lower_both_arms bye
Wake_up look_up look_down bye
Wake_up raise_right_arm move_right_arm_toward_left look_up
open_both_hands bye
Wake_up look_up move_right_arm_toward_left
move_right_arm_toward_right raise_right_arm lower_right_arm
bye
Wake_up raise_both_arms lower_right_arm lower_both_arms
raise_right_arm bye
```

```
Wake_up open_both_hands raise_right_arm close_both_hands
lower_right_arm bye
Wake_up look_up raise_right_arm look_down lower_right_arm bye
Wake_up move_right_arm_toward_left raise_right_arm
move_right_arm_toward_right look_up bye
Wake_up open_both_hands raise_both_arms close_both_hands
lower_both_arms bye
Wake_up open_both_hands raise_right_arm lower_both_arms
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up look_up move_right_arm_toward_left
move_right_arm_toward_right open_both_hands close_both_hands
bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up look_up look_down bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
```

```
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up look_up move_right_arm_toward_left
move_right_arm_toward_right open_both_hands close_both_hands
bye
Wake_up raise_both_arms bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up move_right_arm_toward_left look_up raise_both_arms
open_both_hands bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_right_arm lower_both_arms bye
Wake_up raise_both_arms lower_both_arms raise_right_arm
raise_both_arms bye
Wake_up raise_both_arms open_both_hands lower_right_arm
lower_both_arms close_both_hands bye
Wake_up raise_both_arms open_both_hands lower_right_arm
lower_both_arms close_both_hands bye
Wake_up look_up look_down move_right_arm_toward_left
raise_right_arm lower_both_arms bye
Wake_up look_up raise_right_arm open_both_hands look_down
move_right_arm_toward_left lower_right_arm bye
Wake_up open_both_hands raise_right_arm
move_right_arm_toward_left look_up close_both_hands
lower_both_arms bye
Wake_up look_up look_down bye
Wake_up raise_right_arm move_right_arm_toward_left look_up
open_both_hands bye
Wake_up look_up move_right_arm_toward_left
move_right_arm_toward_right raise_right_arm lower_right_arm
bye
Wake_up raise_both_arms lower_right_arm lower_both_arms
raise_right_arm bye
Wake_up open_both_hands raise_right_arm close_both_hands
lower_right_arm bye
Wake_up look_up raise_right_arm look_down lower_right_arm bye
Wake_up move_right_arm_toward_left raise_right_arm
move_right_arm_toward_right look_up bye
Wake_up open_both_hands raise_both_arms close_both_hands
lower_both_arms bye
Wake_up open_both_hands raise_right_arm lower_both_arms
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
```

```
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up move_right_arm_toward_left look_up raise_both_arms
open_both_hands bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up look_up move_right_arm_toward_left
move_right_arm_toward_right open_both_hands close_both_hands
bye
Wake_up raise_both_arms bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up move_right_arm_toward_left look_up raise_both_arms
open_both_hands bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_right_arm lower_both_arms bye
Wake_up raise_both_arms lower_both_arms raise_right_arm
raise_both_arms bye
Wake_up raise_both_arms open_both_hands lower_right_arm
lower_both_arms close_both_hands bye
```

```
Wake_up raise_both_arms open_both_hands lower_right_arm
lower_both_arms close_both_hands bye
Wake_up look_up look_down move_right_arm_toward_left
raise_right_arm lower_both_arms bye
Wake_up look_up raise_right_arm open_both_hands look_down
move_right_arm_toward_left lower_right_arm bye
Wake_up open_both_hands raise_right_arm
move_right_arm_toward_left look_up close_both_hands
lower_both_arms bye
Wake_up look_up look_down bye
Wake_up raise_right_arm move_right_arm_toward_left look_up
open_both_hands bye
Wake_up look_up move_right_arm_toward_left
move_right_arm_toward_right raise_right_arm lower_right_arm
bye
Wake_up raise_both_arms lower_right_arm lower_both_arms
raise_right_arm bye
Wake_up open_both_hands raise_right_arm close_both_hands
lower_right_arm bye
Wake_up look_up raise_right_arm look_down lower_right_arm bye
Wake_up move_right_arm_toward_left raise_right_arm
move_right_arm_toward_right look_up bye
Wake_up open_both_hands raise_both_arms close_both_hands
lower_both_arms bye
Wake_up open_both_hands raise_right_arm lower_both_arms
move_right_arm_toward_left bye
Wake_up raise_both_arms lower_both_arms raise_both_arms bye
Wake_up open_both_hands close_both_hands open_both_hands
close_both_hands bye
Wake_up look_up look_down bye
Wake_up move_right_arm_toward_left move_right_arm_toward_right
bye
Wake_up raise_right_arm lower_right_arm bye
Wake_up raise_both_arms lower_both_arms open_both_hands bye
Wake_up look_up look_down open_both_hands close_both_hands
raise_both_arms lower_both_arms bye
Wake_up move_right_arm_toward_left raise_right_arm bye
Wake_up move_right_arm_toward_left open_both_hands
raise_right_arm raise_both_arms bye
Wake_up move_right_arm_toward_left look_up raise_both_arms
open_both_hands bye
Wake_up raise_both_arms open_both_hands
move_right_arm_toward_left bye
```