Toward Understanding Human Expression In Human-Robot Interaction

by

William B. Miners

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2006

Abstract

Intelligent devices are quickly becoming necessities to support our activities during both work and play. We are already bound in a symbiotic relationship with these devices. An unfortunate effect of the pervasiveness of intelligent devices is the substantial investment of our time and effort to communicate intent. Even though our increasing reliance on these intelligent devices is inevitable, the limits of conventional methods for devices to perceive human expression hinders communication efficiency. These constraints restrict the usefulness of intelligent devices to support our activities. Our communication time and effort must be minimized to leverage the benefits of intelligent devices and seamlessly integrate them into society. Minimizing the time and effort needed to communicate our intent will allow us to concentrate on tasks in which we excel, including creative thought and problem solving.

An intuitive method to minimize human communication effort with intelligent devices is to take advantage of our existing interpersonal communication experience. Recent advances in speech, hand gesture, and facial expression recognition provide alternate viable modes of communication that are more natural than conventional tactile interfaces. Use of natural human communication eliminates the need to adapt and invest time and effort using less intuitive techniques required for traditional keyboard and mouse based interfaces.

Although the state of the art in natural but isolated modes of communication achieves impressive results, significant hurdles must be conquered before communication with devices in our daily lives will feel natural and effortless. Research has shown that combining information between multiple noise-prone modalities improves accuracy. Leveraging this complementary and redundant content will improve communication robustness and relax current unimodal limitations.

This research presents and evaluates a novel multimodal framework to help reduce the total human effort and time required to communicate with intelligent devices. This reduction is realized by determining human intent using a knowledge-based architecture that combines and leverages conflicting information available across multiple natural communication modes and modalities. The effectiveness of this approach is demonstrated using dynamic hand gestures and simple facial expressions characterizing basic emotions. It is important to note that the framework is not restricted to these two forms of communication. The framework presented in this research provides the flexibility necessary to include additional or alternate modalities and channels of information in future research, including improving the robustness of speech understanding.

The primary contributions of this research include the leveraging of conflicts in a closed-loop multimodal framework, explicit use of uncertainty in knowledge representation and reasoning across multiple modalities, and a flexible approach for leveraging domain specific knowledge to help understand multimodal human expression. Experiments using a manually defined knowledge base demonstrate an improved average accuracy of individual concepts and an improved average accuracy of overall intents when leveraging conflicts as compared to an open-loop approach.

Acknowledgements

> I am a part of all that I have met
> Yet all experience is an arch wherethro'
> Gleams that untravell'd world, whose margin fades
> Forever and forever when I move.
>
> —Alfred, Lord Tennyson

This research and the development of this thesis was an enjoyable and rewarding experience, but not an isolated one. Some of the people that made this journey possible are acknowledged below, but I would like to thank everyone that made this journey an enjoyable one. These helpful individuals include both the named and unnamed, at both work and play. The influence of these people helped to shape this work into what it is today.

I would like to first thank my adviser, Dr. Otman Basir, for providing an excellent environment in which I was able to pursue a research area that interests me, while being able to discuss wide ranging topics, receive valuable short and long-term guidance, endless encouragement, and constructive feedback. He always manages to make every meeting, issue, or discussion enlightening with a positive outcome. I am very grateful for everything that Dr. Basir has done.

I also would like to thank my examination committee, Dr. Mohamed Kamel, Prof. H. Dominic Covvey, and Dr. John Zelek for their investment of time and effort throughout this process. I would like to thank Dr. Hamid Tizhoosh for questions that were especially helpful during the early stages of this research. I would also like to thank Dr. Nicolas D. Georganas for his constructive feedback as my external examiner.

I would like to thank my friends in the PAMI lab and office for the numerous enjoyable discussions, for understanding when the robots lost their way, and for providing valuable help from alternate perspectives, in addition to helpful breaks along the way. There were several different groups that used the robot platform during its initial development. I would like to thank these groups for their patience during development, and for their valuable feedback that was used to help improve the platform.

I would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting this research with a PGS-B scholarship. I would like to also thank my adviser for financial support when needed, in addition to the Faculty of Engineering and the University of Waterloo for scholarships and awards.

My family and friends, both nearby and far away have been patient and supportive throughout this journey. I appreciate their patience, and would like to thank them for their support while I was immersed in this journey. Last, but far from least; my wife, Lucia, and daughter, Natalie, deserve special thanks for their support. Their patience, love, and encouragement throughout this journey was invaluable.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter One

Introduction

If we knew what it was we were doing, it would not be called research, would it?
— Albert Einstein

The ability of a robot to understand human expression is essential for successful natural human-robot interaction to occur. Robots do not currently take full advantage of the information conveyed through multimodal human expressions. Fortunately, multimodal expressions can be understood by associating high-level domain knowledge with low-level perceived elements. Leveraging high-level human knowledge with concurrent elements in a closed-loop system provides a robust and flexible strategy toward designing robots that fully understand the expression of human intent. High-level human knowledge provides precise meaning, whereas low-level machine learned concurrent elements of multimodal human expression provide the precise values required for robot perception. An effective combination of expert knowledge with low-level machine learned elements is required to balance training requirements, flexibility and human maintainability.

Communicating intent to a robot, or, more generally, an intelligent device, currently requires explicit device-friendly instructions that often involve a substantial investment of human time and effort. The primary purpose of these intelligent devices is often overlooked. Almost all intelligent devices are designed to improve human productivity through effective use of time. Ideally, the human mind should be relieved from performing mundane or tedious tasks by offloading this work to nearby pervasive intelligent devices. These devices are not limited to the emerging interactive service robots, or the personal computers that are regularly used for hours, but include devices that perform the less obvious tasks of warming up lunch in the microwave, wireless communication, mail delivery, industrial automation, vehicle control and communication with personal digital assistants. Current research strives to increase the ubiquity of these intelligent devices including the introduction of wearable devices, smart buildings, intelligent vehicles, high-tech appliances, and personal robotic assistants into society.

It is difficult to ignore the benefits provided by the involvement of intelligent devices in our society, but we must also consider the value of our time and the effort required to communicate our intentions. Unfortunately, we spend an increasing portion of the limited time in each day solely communicating with devices. This increasing effort and time required to communicate directly contradicts the goals of intelligent devices to relieve our minds from mundane tasks and improve productivity. If we spend all our time communicating intent, when will our minds be free for creative thought? Our time is valuable, and must not be wasted by slowing down to inefficiently convey intent to poorly accessible devices in our everyday lives.

Inefficiencies in current communication are a direct consequence of conventional tactile input interfaces. Substantial time and effort is required to translate natural thoughts and expressions into a single linear sequence of keystrokes. Although many tactile input interfaces are inexpensive to implement to robustly accept binary inputs, they are not suitable as a sole means for natural interaction. Replacement of these conventional input interfaces with natural and intuitive interfaces will improve the efficiency of communicating intent, and is essential before a device can be truly considered ubiquitous. The substantial effort that we currently exert tediously communicating over inefficient interfaces will be reduced in a natural interface, freeing our minds for more interesting and challenging activities.

When considering the efficiency of an interface, we must not only consider the time spent and effort exerted during communication, but we must also consider the time and effort required for preparation. Preparation includes learning and maintaining a new skill such as touch-typing or a new language such as an assembler. Learning a new language or skill for the sole purpose of communicating intent to devices either requires a person to be fluent and think in multiple languages, or consciously exert effort to translate intent from their native language.

We communicate with one another in a language and manner that feels natural in everyday conversations. Most inter-personal communication tends to be effortless due to our extensive training, continual practise and well adapted perception. Incorporating support for these natural methods in device interfaces will minimize the human time spent and effort exerted to communicate or express intent. The advantages of using our existing, familiar and natural communication methods far outweigh the substantial human effort required to learn and maintain a new language solely to communicate intent to devices. A machine that can understand intent as it is naturally expressed will result in more enjoyable and productive communication experiences.

Several significant steps toward a natural interface have been made, including use of spoken commands, task specific hand gestures, and tactile or haptic input methods with active feedback. Although each of these methods on their own is an improvement over conventional methods, significant progress is required before we can achieve communication that feels effortless. One of the more significant hurdles to overcome is effective use of information contained in multiple modalities and conceptual level knowledge. During inter-personal communication, we understand intent not solely from information provided in any single modality or channel, but from a range of sources simultaneously. Common sources include body language, tone of voice and facial expression in addition to conventional spoken or signed language. The source of meaning in face-to-face communication is cited in widespread literature with only a small portion in the words themselves, and a vast majority in audible channels (i.e. tone of voice) and visible body language [1]. Although this information is frequently generalized and applied out of context [2] it is reasonable to infer that we determine intent using information distributed across a range of channels and modalities.

Use of information distributed across multiple noisy modalities has been shown to reduce recognition uncertainty [3]. Information contained in individual modalities can be incomplete, ambiguous, and even in conflict with the desired intent. Incorporating information from multiple

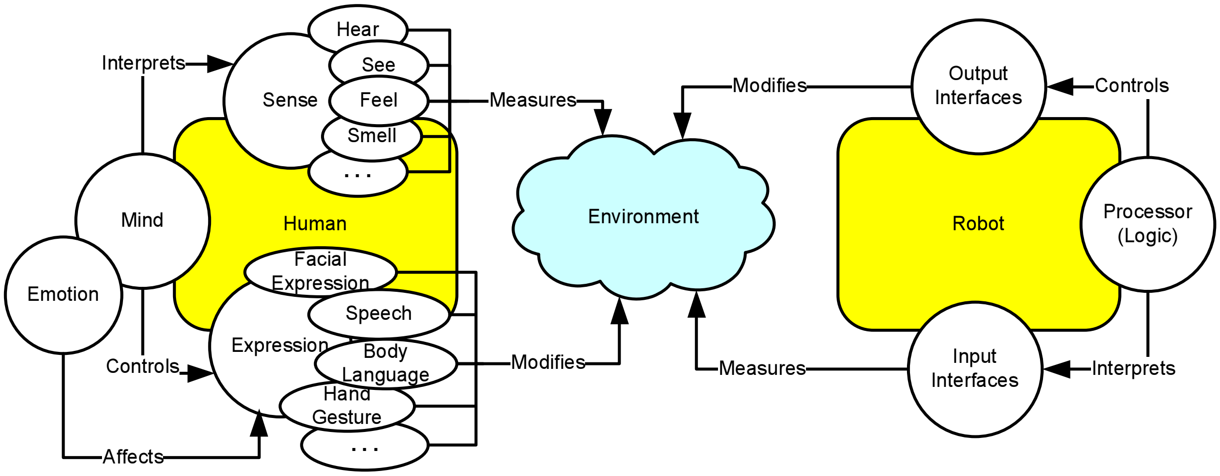


Figure 1.1: The fundamental elements of human-robot communication include a human expression to modify the environment in a manner that a robot can perceive, analyze, and act on. The robot action modifies its shared environment to provide some form of feedback to the human participant.

modalities improves reliability by utilizing the redundant and complementary nature of natural human multimodal communication. Combining this multimodal information is essential to minimize the effort we exert and time we spend communicating with devices in our inevitable ubiquitous computing environments.

1.1 Background

As illustrated in Figure 1.1, multiple channels of communication exist between a human and a robot. At the simplest level, both robots and humans measure their environment through senses or input interfaces and modify their environment through expression or output interfaces. This research focuses on how multiple modalities or channels modify the environment and how this information is measured and used to help understand the original intent being expressed.

Relevant background information for supporting theories and technologies used in multimodal communication is presented in the remainder of this section. This material is classified into conventional interfaces, hand gesture-based communication, facial expressions, and multimodal fusion.

1.1.1 Conventional Interfaces

Effective human-machine interfaces must both accept input and provide useful feedback. Both input and output interfaces to a device must be considered when designing a system to determine human intent. Well designed input interfaces are rendered useless without considering the matching feedback interface. In conventional interfaces, this feedback is often provided visually and human intent is conveyed using tactile input devices. Tactile input devices acquire information about our intent through physical contact. Perhaps the most common tactile input interfaces are the keyboard and mouse pair we use on a daily basis.

The keyboard has changed little from its ancestor, the typewriter, and is still well suited to explicitly dictating words and sentences character by character. Although well suited to word processing tasks, the keyboard inadequately handles natural expression of intent. Almost as common as the keyboard is the mouse or trackball. These pointing devices measure distance and direction travelled over time, allowing us to *point* at locations in a 2D plane. Although we use these interfaces on a daily basis, they are far less intuitive and flexible than a natural finger pointing gesture.

1.1.2 Hand Gesture

In contrast to conventional tactile input interfaces, hand gesture interfaces provide opportunities to convey complex intent without necessarily adopting the syntactical and spatial constraints imposed by a keyboard. This flexibility provides opportunities to express diverse ideas and complex intent. However, the same flexibility has led to a range of different definitions of hand gesture.

Nespoulous defines gesture as “the notion of gesture is to embrace all kinds of instances where an individual engages in movements whose communicative intent is paramount, manifest, and openly acknowledged” [4]. Nespoulous and Kendon define gesture as “The word *gesture* serves as a label for that domain of visible action that participants routinely separate out and treat as governed by an acknowledged communicative intent” [5]. Both of these definitions cover such a wide range of concepts that focused research in gesture fields must further classify and categorize gesture into logical, and manageable, sub-definitions.

Some believe strong parallels between gesture, speech, and handwriting exist and logically lead to classifications of gesture based on speech and handwriting. Others believe gestures are drastically different from language expressed in speech and writing, and classify gestures independently from speech and handwriting [6]. A third group of researchers believes that gestures are primarily extensions of speech [7]. All these viewpoints of hand gesture classifications are valid, and help to define sub-classifications of the more general hand gesture definitions. Hand gestures formed as extensions of speech are commonly referred to as *gesticulation* and gestures formed independent from speech are referred to as *autonomous gestures.* This type of classification of gesture is illustrated in a scale introduced in by Kendon and later extended by McNeill is shown in Figure 1.2 [6]. In this classification, the requirement of speech decreases in each class from left to right, or the independence from speech increases from left to right. Additional material about classifications of hand gestures can be found in Appendix B. Throughout this thesis, the author applies a broad definition of gesture to encompass both hand gestures and facial expressions.

Gesticulation → Language-like Gestures → Pantomimes → Emblems → Sign Languages

Figure 1.2: Kendon’s Classifications of Hand Gesture

> A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. All that matters is which key was pressed. [8]

Although McNeill concentrates on gesticulation, the importance of distinguishing between hand gesture and other forms of language is emphasized. Spoken and written language can be used to represent meaning by dividing the meaning up into a series of words (temporal segmentation) and organizing each of these words in a hierarchical manner into sentences. Spoken and written language is restricted to change in one dimension, time. McNeill emphasizes that gestures are never hierarchical, as two gestures produced together do not combine to form a longer higher-level gesture (noncombinatoric) as two words might. Another significant difference between gestures and speech or written language is that the meanings of individual gestures are determined by the complete gesture *phrase*. One gesture often conveys multiple meanings simultaneously, whereas in speech or written language, words are analyzed and combined to extract meaning.

1.1.3 Facial Expression

As humans, we can effortlessly determine the intent of facial expressions as early as infancy. Unfortunately, automating this process for use in a human-machine interface is a substantial challenge. Facial expression encompasses a wide range of communication channels, including dynamic properties; lip motion, eye gaze and emotion and static properties; gender, age, and perhaps even personality. The non-rigid nature and subtleties present in facial expression complicate recognition and understanding of intent.

In contrast to face recognition where individuals are identified using their facial characteristics, facial expression recognition does not discriminate between individuals. Facial expressions primarily convey emotion and can be characterized by several different features. Common features include the eyebrows, eyes, nose, mouth and chin. Similar to hand gestures, these features can be analyzed as a whole, as a set of individual subfeatures, or at some compromise between these two extremes.

Facial expressions are often described using formally defined categories. The most common description method is the facial action coding system (FACS), originally designed for human annotation of facial expression. This system breaks facial expressions down into 44 individual visual changes, each defined as an action unit (AU).

Before emotion can be determined from facial expression, emotions themselves must be defined. Unfortunately, quantifying emotion is a nontrivial task. Six commonly used categories of emotion are happiness, sadness, surprise, fear, anger, and disgust [9]. Although these categories are frequently challenged, they remain the most commonly used in current literature.

1.1.4 Multi-Modal Communication

The source of meaning in face-to-face communication is cited literature as approximately 7 percent in the words themselves, 38 percent in complementary audible channels including tone of voice, and 55 percent in visible body language [1]. Although these statistics are frequently generalized and applied out of context [2] it is still reasonable to infer that we determine intent using information distributed across a range of sources. [1]

It is important to first clearly define the term *multimodal* before we continue. One definition is provided in [10], where multimodal is simply more than one modality or mode. Each physical communication method is considered a modality, whereas a mode describes the context of the communication. Modalities can also refer to the way an idea or action is expressed or perceived [11]. Sound can be considered one modality as it is perceived by hearing, and vision another modality as it is perceived by sight.

The sources of intent in human face-to-face communication are expressed across not only multiple modalities but also multiple channels [2]. Information in each modality can be split across or duplicated in more than one channel, where each channel encodes a logical component of intent. A communication channel is simply a connection between two physical or abstract locations. Information contained in each channel is not necessarily independent, as these channels may include a visual channel for facial expression, a tactile channel for hand gestures, and separate visual channels for hand gestures and lip motion.

In our research, the type of perception used to sense information defines a modality, including vision, hearing, touch, and smell. Facial expressions are normally perceived using the single modality of vision, whereas hand gestures can be perceived using vision or touch. Speech information is also conveyed across multiple modalities including vision and hearing. Fusion of this human communication expressed across different modalities and channels is a big challenge that must be overcome before human-robot communication will feel as natural as interpersonal communication.

1.2 Objectives

> A loud clatter of gunk music flooded through the Heart of Gold cabin as Zaphod searched the sub-etha radio wave bands for news of himself. The machine was rather difficult to operate. For years radios had been operated by means of pressing buttons and turning dials; then as the technology became more sophisticated the controls were made touch-sensitive – you merely had to brush the panels with your fingers; now all you had to do was wave your hand in the general direction of the components and hope. It saved a lot of muscular expenditure, of course, but meant that you had to sit infuriatingly still if you wanted to keep listening to the same program.
>
> — Douglas Adams, The Hitchhiker's Guide to the Galaxy

Although many of today's user interfaces are not as restrictive as those in the Heart of Gold cabin from the Hitchhiker's Guide to the Galaxy, the interfaces currently available still force their users to communicate using unnatural methods. To relax the unnatural constraints imposed with unimodal, single channel communication, this research focuses on the design of an architecture

[1] The study that resulted in these often misused statistics was based on three individuals speaking the single word *maybe* and 17 judges determining intent based on visual and audible information with no feedback.

and supporting algorithms to effectively express intent with intelligent devices using information distributed across multiple sources. The primary objectives of this research are:

1. To design a human-in-the-loop architecture for flexible multimodal natural human-robot communication. The closed-loop nature of this architecture provides support for iterative refinement of intent.

2. To design and evaluate an architecture to integrate information between multiple modalities and leverage human expertise at appropriate stages in the understanding process.

3. To evaluate the effectiveness of integrating facial expression and hand gesture data to understand human intent in a specific domain.

4. To adapt a knowledge representation and reasoning technique to appropriately handle uncertainties in both relationships within the knowledge base and between the knowledge base and descriptions of multimodal human expression. Although several formal knowledge representation and reasoning methods exist, few provide adequate support for spatio-temporal uncertainties and cross-modal correlations.

5. To evaluate the impact of analyzing conflicts and providing feedback from high-level intent to lower-level hand gesture and facial expression components of the understanding process.

1.3 Organization of Thesis

This thesis is organized to address the identified objectives and present the research as it relates to the flow of information from human conceptualization and physical expression through to device perception, understanding, and analysis of conflict. This bottom-up organizational approach is divided into four distinct chapters corresponding to the areas of focus identified in Figure 1.3. The motivation for the author's work was presented in this chapter, and is followed by a review of the state of the art in multimodal human-robot interaction. The author's approach is introduced in Chapter 3, starting with the robot's perception of human expression and leading to the recognition of basic elements of hand gestures and facial expressions. The following chapter presents the approach and techniques used to understand and form hypotheses of human intent, succeeded by a discussion of leveraging conflict in Chapter 5. An implementation of this approach is presented in Chapter 6, including the experimental design used to empirically confirm the validity of the approach. Conclusions are presented in Chapter 7, including a discussion of insights and questions raised throughout this work and future directions that will be valuable toward achieving seamless integration of interactive robots in human society.

The organization of this thesis can also be considered from the perspective of Bloom's taxonomy. Chapter 2 analyzes the state of the art in human-robot interaction to identify limitations and issues that must be resolved before robots can fully understand human expression. The synthesis of the author's architecture and approach is presented in Chapter 3 through Chapter 5.

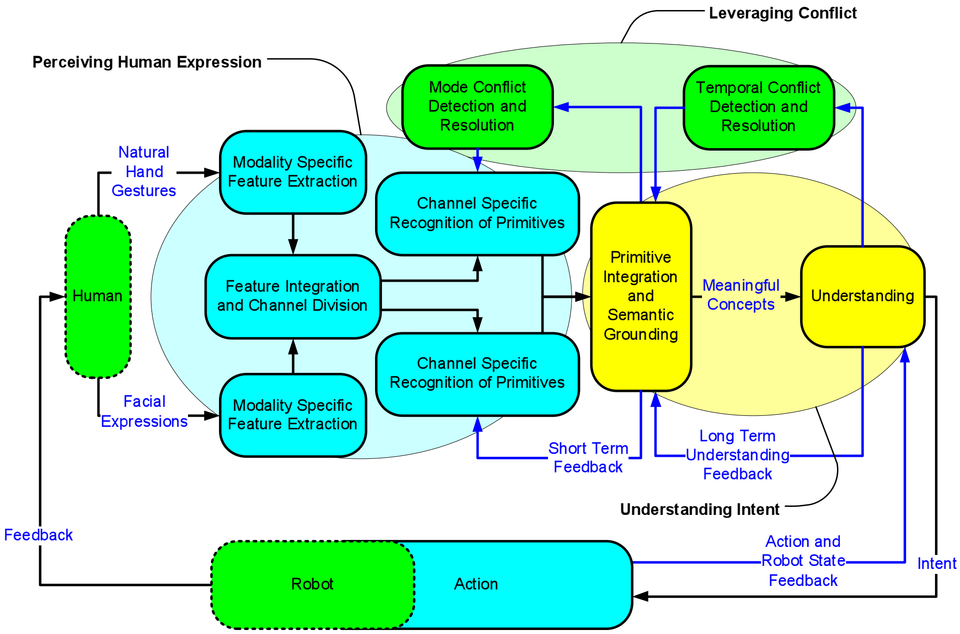

Leveraging Conflict
Perceiving Human Expression
Mode Conflict Detection and Resolution
Temporal Conflict Detection and Resolution
Natural Hand Gestures
Modality Specific Feature Extraction
Channel Specific Recognition of Primitives
Human
Feature Integration and Channel Division
Primitive Integration and Semantic Grounding
Meaningful Concepts
Understanding
Channel Specific Recognition of Primitives
Facial Expressions
Modality Specific Feature Extraction
Short Term Feedback
Long Term Understanding Feedback
Understanding Intent
Feedback
Action and Robot State Feedback
Intent
Robot
Action

Figure 1.3: The presentation of this thesis follows the flow of information from human expression through to robot understanding. This approach is divided into three major areas; the perception of human expression, robot based understanding, and leveraging conflict for iterative refinement.

An evaluation of the approach is detailed in Chapter 6, with conclusions drawn and a selection of valuable future directions identified in Chapter 7.

Chapter Two

Human-Robot Interaction

A review of the state of the art in perceiving and understanding hand gesture, facial expression, and multimodal human expression is provided in this chapter. Techniques and approaches are categorized by their purpose: feature extraction and representation, recognition, multimodal fusion, and understanding. Almost all current literature is based on a sequential series of independent steps that correspond to a subset of these categories.

2.1 Feature Extraction and Representation

The selection and representation of features in a robot's environment determines the information available for further processing. Appropriately selected features should represent information required to understand human expression with a minimum of imposed restrictions. The modality used for perception often influences the approach, as does the purpose of this information. Vision is one of the most popular methods of perception, although wearable sensors often provide more accurate information. Two primary feature extraction and representation methods exist for human expressions conveyed in hand gestures and facial expressions: appearance-based and model-based.

2.1.1 Appearance-Based

Appearance based techniques are generally less computationally expensive than model based techniques since a conversion from appearance space to model space is not required. Features are selected directly from 2D information, without requiring a translation into 3D as might be required in model-based techniques. The use of this 2D information lends itself well to perception using vision.

Appearance-based techniques can be further divided into those that use features or points of interest, those that use dimensionality reduction techniques directly (i.e. eigenvectors), and those that use statistical properties of the image without feature identification or correspondence.

2.1.1.1 Statistical Properties of Image

Use of histograms of local operators is common in appearance based recognition techniques, including use of local orientation information to recognize static poses [12], use of colour [13], or combinations of features [14]. The approach presented in [14] uses a generalized framework of multidimensional histograms of local receptive field operators to recognize objects probabilistically without segmentation. This probabilistic method has a low computational complexity and has been experimentally shown to perform well at recognizing a variety of different general objects from the Columbia image database (COIL-20). Unfortunately, histograms of local operators cannot distinguishing between different hand configurations or facial expressions well due to the close similarity in their overall appearance.

In [12] hand gesture recognition is approached from a speed / performance standpoint, where use of an appearance-based method is a reasonable compromise between accuracy of data and speed of computation. Their method is able to distinguish between approximately 10 different hand gestures, which might be sufficient for specialized interfaces, but the ability to distinguish between a larger number of hand gestures is required for natural human-robot interaction.

Although appearance based recognition techniques can be easily applied to recognize individual images, or static poses, these techniques do not lend themselves well to incorporating temporal information from gestures. One method for extending these appearance-based techniques into the temporal domain is to consider an entire sequence of images during recognition. In [15], a method of this type is proposed to capture dynamic information in gestures. Darrell and Pentland used dynamic time warping (DTW), a simplification of a hidden Markov model (HMM) approach to compare sequences of images against previously trained sequences by adjusting the length of the sequences appropriately.

2.1.1.2 Feature Based

Feature based recognition techniques are based on extracting image features such as corners or significant edges and matching the features of the known representation with the observed features. Some researchers extend this technique by grouping several image features together to produce high-level object features to improve recognition accuracy. This technique of matching the image features, or identifying the correspondence between the known object and the observed object is computationally demanding and increases exponentially with the number of features used.

A common feature-based technique is elastic graph matching. Elastic graph matching involves representing objects as labelled graphs with nodes containing local image information and edges containing geometrical information. One graph is used to represent each object and the algorithm uses this graph to scan across an image during recognition in an attempt to match each node and edge with the image. The *elastic* portion of elastic graph matching occurs during recognition when the graph is allowed to change shape, as a result of scaling, rotation, or stretching.

It is possible to achieve high recognition rates of hand gestures in a controlled environment, but these rates are very low in complex, cluttered backgrounds. The problem of pose recognition against complex backgrounds is addressed in [16] by using an elastic graph matching technique for hand gestures. Bunch graphs of jets based on a wavelet transform of Gabor-based kernels are used to represent known poses. A classification rate of 86.2% is achieved against complex backgrounds with different users, but its drawbacks include manual positioning of graph nodes during training to generate the models and the computational complexity of matching each graph against an unknown image. Using a supplemental hand localization technique or rough segmentation will improve the performance of the method proposed by Triesch and Malsburg.

This same technique is applied to hand gesture recognition in [17] but is limited to six static postures in a finger spelling alphabet. The major disadvantage is the computational complexity of the algorithm. The relatively small training set required for high recognition rates can be considered an advantage over other techniques such as neural networks, but since graph nodes are positioned manually, it is unfair to compare this against more automatic training techniques.

The inconvenience of manual positioning required in [16, 17] is addressed in [18] in a face recognition application. The technique proposed in their paper performs well only when recognizing objects of the same class (*in-class recognition*), restricting its use to pre-classified and segmented objects. Techniques used to recognize individuals using their face appearance [18] are very similar to techniques used to recognize facial expressions. Two major appearance-based feature extraction approaches exist in current literature: High-level whole-face facial expressions based on known emotional states and low-level actions or features corresponding to specific portions of the face.

Dynamic properties of edges of the mouth, eyes, and eyebrows have been used to classify six prototypical emotions [19]. These features allow emotion to be inferred using cues of motion near to the *peak* static pose of a specific facial expression.

Approaches that extract features from individual parts of the face rely primarily on facial action units defined in FACS [20]. These features often result in a more flexible interpretation of facial expression than the whole-face approaches. Since the existing FACS system is well suited for manual annotation of facial expression, research has attempted to use a subset of the same annotation system but automate the process. In [21], six action units corresponding to the eyes and eyebrows were selected as the features to extract. An approach limited to head movement uses the area between the eyes to track head motion, recognizing nods and shakes [22].

2.1.1.3 Principal Component Analysis

Images can be efficiently represented by breaking them down into their principal components, resulting in a small number of coefficients, however, any small change in the image changes these values. This global technique is sensitive to partial occlusion, translation, rotation, changes in illumination, and scale, limiting the technique's applicability to specialized tasks under specific constraints. One task where eigenvectors are applicable and commonly used is the recognition of faces, where the faces are pre-segmented and normalized before the eigenvectors are calculated. Eigenvectors are used in [23] to represent the most discriminating features in a combination of fovea images of the hand, local motion, and global motion.

Moghaddam and Pentland use eigenspaces and principal component analysis (PCA) not only to extract features, but as a method to estimate complete density functions for localization in [24]. A set of images can be represented in a very small amount of data by using PCA. The Karhunen-Loeve Transform is used to identify representative eigenvectors, from which PCA is used to break the high-dimensional eigenspace into lower-dimensional orthogonal subspaces. Each image can then be represented in terms of its co-ordinates in eigenspace, or its deviation from the *average* image. This technique is used to recognize edge-maps of pre-segmented hands with a high success rate, but as with most eigenvector approaches, the representation in eigenspace is highly susceptible to noise and requires very careful localization and segmentation to function properly.

A maximum likelihood technique for recognizing 2D images in a database of eigenspaces is presented in [24]. This technique is combined with the continuous HMM approach in [25] to recognize sequences of eigenspace representations over time. A similar approach is presented in [26]. However, an eigenspace representation of the optical flow is used to model motion rather than an eigenspace representation of the static spatial information. Eigenvectors are also used in [27] to represent images for recognition.

Typically, the emphasis in recognition is on the spatial, or pose information. Hand gesture training is approached in [23] with an equal emphasis on motion (temporal) and spatial information. Global (i.e. movement of the hand), local motion (i.e. change in shape of the hand), and fovea images of the hand are decomposed into eigenvectors and used as features to recognize hand gestures using a space-partition tree. This approach is unobtrusive and has a logarithmic complexity during recognition, but it may be challenging to account for the co-articulation that occurs in continuous hand gesture using this technique. Since the approach is global, every unique hand gesture must be learnt and recognized, even if the only difference between gestures is the position of one finger.

The eigenspace technique proposed in [24] is utilized in [25] for representation and detection in space and adapted to combine this technique with the continuous HMM technique for recognition of sequences of eigenspace representations over time. A similar approach is presented in [26], however, an eigenspace representation of the optical flow is used to model motion rather than the static spatial information. Facial features are also represented with some success using eigenvectors (or *eigenfaces*) [28].

2.1.1.4 Optical Flow

Optical flow features are used to recognize specific facial expressions by the movement of facial components against a stationary head with high rates of recognition for the six basic emotions approaching 80 to 94% [19, 29].

2.1.2 Model-Based

Model based techniques require a transformation between the acquired 2D image and a 3D model of the known hand configuration or facial expression. This transformation is generally more complex than appearance-based techniques, but can result in reduced storage requirements. In appearance-based techniques, each unique viewpoint of the hand or face must be represented uniquely, whereas in model-based techniques, the underlying 3D model is represented once and can be matched with a number of different viewpoints.

The issue of 3D hand gesture analysis using a single vision source and bio-mechanical constraints is addressed in [30]. The predicted hand model is projected on to the 2D vision source and adjusted using several optimization techniques to minimize differences. A similar model-based technique is used in [31] where a single vision source is used to determine 3D hand pose and resulting joint angles using neural networks. A 22-sensor CyberGlove is used to provide training data, and tests demonstrated that the 3D hand pose can be reconstructed in a viewpoint invariant manner.

Although the entire hand is not modelled, a model including the position and orientation of one finger and the thumb are determined using a stereo vision technique in [32]. A more detailed approach is presented in [33], also using stereo vision to reconstruct a 3D hand model. 3D articulated models of not only the hand, but the entire body are reconstructed from silhouettes in [34].

A *synthetic muscle* approach is currently the most realistic method to generate facial expressions, and is used in model-based facial expression research [35]. These approaches build a facial model using generated muscles to pull portions of the skin, but are often based on ad-hoc muscle placement due to the complexity of accurately modelling real muscle positions and their physical properties [36].

Although the majority of hand gesture and facial expression research relies on visual information, alternatives are available. Using physical sensors to determine joint angles or physical locations of body parts reduces the ambiguity and improves the accuracy of the description of body parts when compared against vision-based techniques. The major disadvantage of physical sensors or markers is their obtrusive nature, limiting their usefulness in practical everyday human device interfaces.

Physical measurement of eyebrow features was attempted with limited acceptance [37]. Data gloves are more frequently used and have been combined with vision systems to translate between voice and sign language and vice-versa[38]. Some of the outstanding problems include segmentation of continuous gestures, as this system required the operator to pause between each sign. Since no logic is used during interpretation, sentences are produced that do not make sense when the system incorrectly recognizes one or more words or hand gestures.

Gloves are also used in [39], but not for direct physical measurement. Gloves are used in [39] to provide reliable individual hand colour cues for use in a vision-based system. Whether the glove is used for physical data acquisition or used to modify properties of the hand, it is still an inconvenience to the user. Starner and Pentland observe a 99% individual word recognition rate when the coloured gloves are used and a 92% recognition rate without the gloves.

Magnetic trackers are used in [40] to determine arm joint angles and recognize planar arm gestures. Human arm dynamics are modelled based on knowledge of these arm joint angles, and the closest match between multiple Kalman filters is used to discriminate between four different arm gestures. Similar approaches use infra-red high-speed trackers or 3D laser scanners to precisely measure facial expressions.

2.2 Recognition

Once features are extracted and appropriately represented, a variety of techniques are applied to recognize hand gestures or facial expressions in current literature. Recognition techniques follow one of two major categories: continuous or discrete. Continuous techniques explicitly handle time in a stream of information, whereas discrete techniques are applied to static or temporally-segmented components. Recognition of static hand poses can achieve high recognition rates, but when recognizing hand gestures in continuous sequences, a means to appropriately segment the stream is required. In addition, adjacent gestures may affect one another as adjacent words cause co-articulation in speech. In facial expression recognition, considerable information is encoded in the motion of facial features. Recognizing a facial expression using a static pose produces poor results without use of a priori knowledge.

The approach of training individual gestures by example and subsequent recognition is considered inappropriate in gesticulation [27]. In gesticulation, the spatial information may not be as significant in conveying meaning as the temporal information in the gesture.

2.2.1 Hidden Markov Models

The HMM provides a stochastic framework that is commonly used for recognition, both to implicitly segment gestures from one another and to recognize pre-segmented gestures. In a standard HMM, a sequence of events is recognized through the movement between hidden states based on a given sequence of observations. Each transition between states has a property defining the probability that the system will move along the transition given the system was in the preceding state. In a continuous HMM, each state has a probability distribution function associated with it to describe when the system is really in each given state. Common models include the left-right model illustrated in Figure 2.1. This model handles non-cyclic hand gestures with varying durations in each state. The loop-back transitions allow the duration of each state of the gesture to vary, and the skip transitions allow portions of the gesture to be omitted (dynamic time warping). Please refer to Appendix A for further HMM details.

The HMM is used extensively in sign language recognition, including recognition of whole word signs in a 40-word American sign language (ASL) vocabulary [39]. Hand velocities and orientations in two dimensions are used as observations, limiting the system to signs that are very different from one another. Since individual finger information is not determined, it would be challenging to differentiate between similar signs using these observations. Early approaches imposed a restrictive sentence form of *pronoun, verb, noun, adjective, pronoun* to reduce the search space for each gesture based on its position in the sequence, and to assist in segmentation [39].

Liang and Ming determine when to segment gestures based on the highest probability of two adjacent gestures [41]. The division between two adjacent gestures is adjusted (three different positions are used), and a HMM is used to recognize each gesture before and after the division. The pair of gestures with the highest likelihood are then selected. Although this technique is common, it does not take co-articulation into account and increases the computational complexity linearly with the number of divisions evaluated. Liang and Ming's approach lends itself well to

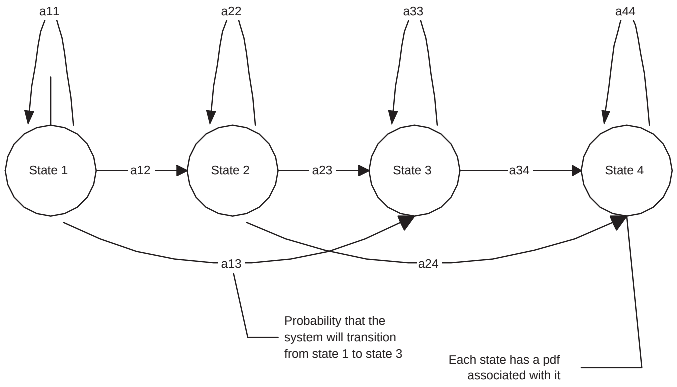

a11
a22
a33
a44
State 1
a12
State 2
a23
State 3
a34
State 4
a13
a24
Probability that the system will transition from state 1 to state 3
Each state has a pdf associated with it

Figure 2.1: Left-Right Hidden Markov Model With Skip States

the definition of gesture as a sequence of static poses, as the division points can be selected between discrete hand poses.

A simple method to segment hand poses relies on a time varying parameter (TVP). This method can be used to detect when a significant pose exists by using a threshold value on motion parameters. Although this approach may work well in Taiwanese sign language (TSL), potentially significant motion information is discarded. Motion information has been incorporated into the same TVP model, but only as a descriptive feature of the segmented static poses [41].

Vogler and Metaxas approach the co-articulation problem in the scope of ASL two different ways, using a context dependent HMM to implicitly handle co-articulation and using the relatively static structure of ASL to explicitly model the effects of co-articulation with a separate HMM for each permutation [42]. It is shown that explicitly handling the effects of co-articulation is more effective than the use of a context dependent HMM, and that in ASL, *co-articulation* may either delete a hold and/or add a movement rather than changing the sign itself as words are changed from co-articulation in speech.

A HMM is permitted to adapt on-line by Wilson and Bobick in [43] rather than explicitly separate training and recognition processes. The amount of adaptation must be carefully controlled to prevent similar gestures from becoming confused. Wilson and Bobick's implementation is restricted to a single beat gesture, avoiding the issue related to adaptation of similar gestures.

The coupled hidden Markov model (CHMM) is introduced in [44] to model multiple interacting processes while maintaining the Markov condition that each state must depend only on the prior state. It is shown that the CHMM can recognize complex sequences with higher recognition rates than either the linked HMM or the traditional HMM. Please refer to Figure 2.2 for an illustration of the differences between the traditional, linked, and coupled HMM (skip states and loops omitted for clarity) as presented in [44].

Iwai et al. also believe that the Markov condition is insufficient to appropriately represent gesture [26]. Their approach does not use the CHMM, but includes automaton to adjust the output probabilities in each HMM based on the current and previous gesture. This allows the automaton

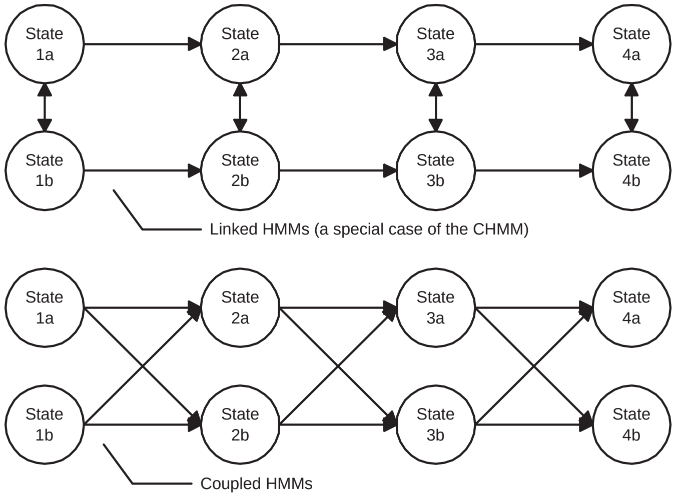


Figure 2.2: Coupled Hidden Markov Model

to model dependencies across longer time intervals (between gestures) while each HMM models shorter time intervals (within gestures). This technique is shown to improve recognition over a simple HMM in complex gesture sequences, but the usefulness of the automata is dependent on the accuracy of the gesture recognition results from the HMM.

Vogler and Metaxas emphasize the difficulty in scaling the CHMM and HMM to large vocabularies [45]. Their approach models gesture by recognizing lower level components in ASL with a parallel hidden Markov model (PaHMM). This allows a small set of HMM to be used in parallel to represent a large portion of the ASL. Results from using the PaHMM are obtained by temporally segmenting and combining at the “word” level. Use of interpretation information for segmentation and allowing each parallel HMM to overlap in time may allow this technique to be applied to gesture interpretation (as opposed to sign language recognition).

The potential drawbacks of using the HMM include the assumption that successive observations are independent and that the probability of being in any given state depends only on the prior state [46]. This restriction may not be sufficient for complex gestures, and has been identified as an inappropriate model for complex sequences in [26, 44]. In addition, an appropriate model for the process must be developed ahead of time and the observations must be segmental or divisible in order for the HMM to perform well.

Issues related to training and recognizing gestures on-line have been tackled by iteratively adapting the parameters of the HMM [47]. This research uses a CyberGlove to eliminate the problem of feature selection, and uses Bakis topologies where the system may move from one state to itself or to one of the next two states, which describes simple, non-cyclical gestures well. Although these methods do not allow for training of completely new gestures, they do provide means for robots to adapt to minor modifications in gestures. Their approach is considered discrete since the operator is required to be still between gestures.

Traditional recognition using the HMM has been adapted to include the sensitivity of overall gesture information to specific a parameter [48]. This allows a gesture such as *pointing* to be recognized using a HMM with the parameter of the direction of the finger being modelled explicitly. In this approach, Wilson and Bobick present a method to interpret parameterized gestures, where a parameter such as orientation or movement is considered along with the pose. Their approach adapts the output probabilities in the HMM's based on the parameter(s), and is applied to gesticulation examples where key words such as "I caught a fish. It was *this* big" can be used to trigger the interpretation of gesture. The extraction of appropriate features is identified as an important problem, but independent from recognition and interpretation. One of the drawbacks to Wilson and Bobick's technique is the requirement to explicitly specify the parameter(s) during training.

2.2.2 General Finite State-Based

Automata are used in [26] to model dependencies over long durations, while using a HMM to model the shorter time intervals. The automata are used to update output probabilities of each HMM. This new technique is shown to improve recognition over a simple HMM in complex gesture sequences, but its usefulness depends on the accuracy of each underlying HMM.

Other state based approaches include building finite state machines from sampled data offline by handling the spatial information first, then incorporating temporal information [49]. Recognition and progress from one state to a subsequent state is based on threshold values that are calculated during training. Manually generated finite state machines are also used to model temporal relationships in multimodal communication [50].

2.2.3 Neural Networks

A common implementation of neural networks in gesture recognition is the time-delay artificial neural network (TDNN). A TDNN has several layers to represent relationships in time between its inputs. It has been shown that TDNN's can be used to classify 2D motion patterns with a high recognition rate using position and velocity inputs relative to the head of the person [51]. The TDNN observes only a small portion of the input at a time, making decisions on each portion before moving on to the next portion of the input. Once the entire input is processed, the series of smaller output decisions are combined to a single decision, the class of a gesture. Each motion pattern is learnt by a TDNN using the standard error back propagation learning algorithm, and as with all neural networks, a relatively large volume of training data is required.

Neural networks have also been applied in fingertip detection in [52]. Gabor filters are used on grey-scale low-resolution (80x80) images as inputs to the networks. Separate networks are used to identify individual digits. Similar vision information is used in [53] for input to neural networks, but rather than using Gabor filters on the original image, multi-scale representations are used to identify key objects by colour, structure and motion. Their gesture interpretation approach is limited to static gestures that remain static for a predefined period of time and the task of representing and actually interpreting the gesture is left as an opportunity for future work.

Additional approaches using neural networks include identification of the target position of deictic pointing gestures [54] and use of multilayer neural networks to translate gestures to speech [55, 56]. Support vector machines have been used to recognize combinations of upper facial action units given a static head position with an average AU recognition accuracy of 61% on a user-independent dataset [21].

2.2.4 Elastic Graph Matching

Elastic graph matching involves representing objects as labelled graphs with nodes containing local image information and edges containing geometrical information. The problem of static hand pose recognition against complex backgrounds has been addressed by using an elastic graph matching technique [16]. Bunch graphs of jets based on a wavelet transform of Gabor-based kernels are used to represent known poses with a classification rate of 86.2%. Drawbacks include manual positioning of graph nodes during training and the high complexity of the recognition process. This same technique has been applied to recognize six static postures in a finger spelling alphabet [17]. The relatively small training set required for high recognition rates can be considered an advantage over other techniques such as neural networks, but since graph nodes are positioned manually, it is difficult to compare this against more automatic training techniques. Some of these inconveniences were addressed in in a face recognition application [18]. Unfortunately, the face recognition application performed well only when recognizing objects of the same class (*in-class recognition*), restricting its use to pre-classified and segmented objects.

2.3 Multimodal Fusion

Although considerable research exists concentrating on recognition of information conveyed across single channels and single modalities, little research uses the redundant and complementary information contained across multiple channels and modalities during natural communication. Some of the first multimodal work includes the popular *put-that-there* research that substituted speech keywords with deictic gesture [57]. More than two decades later, a substantial portion of current multimodal research continues to use this simple keyword substitution approach [58–60], although some exceptions exist including research that combines emotion recognition using facial expression with audible speech [61, 62].

An application for keyword substitution was identified in map interfaces [63]. This application follows directly from the *put-that-there* research, extracting deictic hand postures based on the timing of keywords. More flexible approaches incorporate syntactical relationships between modalities in formal grammars [50]. These approaches still substitute information in one modality with another without feedback or mutual disambiguation.

Recent research has proposed fusion of gesticulation and speech based on the catchment model [64]. This recently proposed approach concentrates on the relationships between discourse and whole hand motion in an intuitive manner, but does not consider mutual disambiguation. Fusion of data between multiple modalities has clearly been shown to improve results in a diverse range of fields [65]. Interest in multimodal communication is increasing, and ample opportunity exists to use information from multiple channels or modalities to improve reliability and accuracy [66, 67].

2.4 Interpreting and Understanding

Interpretation techniques deal with the problem of combining a sequence of symbols in a useful manner. The techniques are commonly divided into syntax (rules of symbol combination) and semantics (meaning), although the division between these two is never distinct.

2.4.1 Syntax

The emphasis in current hand gesture literature is on the recognition process, with little emphasis on syntax. Word-level syntactical analysis is often performed by providing template sentences in which the set of individually recognized gestures representing words are matched against without regard for position [38]. A threshold value is used to determine if the sentence adheres to the provided syntax or not. This approach, although simple, restricts use of words or symbols to the predefined template sentences.

Strict constraints are sometimes imposed on the language structure to improve recognition rates [39]. Each hand gesture *sentence* in this research is composed of exactly five words, a personal pronoun, verb, noun, adjective, and the same personal pronoun. A vision-based skin-colour tracking system is used to extract hand features for use in gesture recognition using a HMM approach and it is shown that the strict grammar improves the accuracy of individual gesture recognition over a standalone HMM approach by 7%.

Using a constrained gesture language in this manner not only simplifies the process of synthesis, but reduces the complexity of the recognition process since the number of possible gesture models are restricted based on the position of the gesture in the constrained language structure. In natural human-machine interaction, it might not be possible to enforce such rigid constraints on the gesture language itself, as the strict rule-based grammar does not easily extend to account for variations in phrase composition or length.

A rule-based action recognition system is presented in [68], in which sequential hand movements are constrained by a set of rules, and individual components are recognized using the 3-state Bakis HMM. This approach utilizes whole-hand motion information, and reduces the complexity of overall recognition by breaking hand movements down into smaller components, but relies on these smaller components being purely sequential.

Perhaps one of the most popular techniques, stochastic models can be used for both segmentation and for syntactic analysis. Segmentation points can be determined using the probability of occurrence between adjacent gestures [69]. Statistical language models are also used in [42] to improve recognition accuracy in the same manner that statistical language models are used in speech recognition. Both of these researchers use bi-gram models to reduce the complexity of the language structure, but only with small sets of gestures. Statistical language models are not always applicable when any of the following properties are true [68]:

- Complete data sets are not always available, but smaller examples can easily be found
- Semantically equivalent processes possess radically different statistical properties
- Structure of the process is difficult to learn but is explicit and a priori known

Bobick and Ivanov apply manually generated stochastic context-free grammars to reduce the ambiguity of individual actions and to provide a priori information to the lower level action detection process [68]. Although their approach is limited to temporal information, it should be possible to extend this technique to include the spatial information required for many gestures. Some of the disadvantages of the stochastic context-free grammar technique used by Bobick and Ivanov include the difficulty in manually generating an accurate grammar model with probabilities.

2.4.2 Semantics

Several possible approaches to the interpretation of motion in general are outlined in [70]. This paper divides the interpretation of motion into three groups based on type of activity: movements requiring no context or history, activities that are based on probabilities and a history of movements, and actions that describe meaning. It is emphasized that most research in recognition is in the area of recognizing activities, or sequences of motions, and this research does not address the higher level task of interpreting action.

It is proposed that meaning is fundamental to the definition of hand gesture and a new categorization method for hand gestures is introduced by Hummels and Stappers [71]. Their research is in the area of gestural product design without using speech, emphasizing the end user of the gesture interface rather than the technical aspects. This perspective is important to draw attention to the fact that the existing *static-posture* recognition systems are sometimes as difficult to learn and use as a command line interface for end users. Hummels and Stappers believe gestures refer to space, pathetic information, symbols and emotion simultaneously. Although the experimental approach used a *wizard-of-oz* approach (a person controlling the machine feedback to the user) to simulate a gesture recognition interface, the emphasis on meaning in gesture is important in gesture interpretation. It is important to note that an algorithm or recognition technique is not presented by Hummels and Stappers, but a proposal for an alternative emphasis on gesture interpretation.

The importance of including logical constraints in human action recognition systems has been emphasized with the use of Allen's interval algebra past-now-future (PNF) networks to constrain actions [72]. Although cyclic actions are not supported, and the framework for logical constraints is extremely simplified, the importance of logical constraints can be applied directly to gesture interpretation to ensure the interpreted meaning is reasonable. Pinhanez and Bobick also introduce how PNF networks can be used to introduce some fault-tolerance into human action recognition systems by eliminating *impossible* actions using the logical constraints.

A method to interpret parameterized gestures is introduced in [48], where a parameter such as orientation or movement is considered along with the pose. This approach adapts HMM's to explicitly model the parameters, and is applied to gesticulation examples where key words such as "I caught a fish. It was *this* big" can be used to trigger the interpretation of gesture. The extraction of appropriate features is identified as an important problem, but independent from recognition and interpretation. One of the drawbacks to this technique is the increased training complexity as each parameter must be explicitly specified during training.

Conceptual dependencies are introduced in [73] and later applied in a multimodal medical virtual reality context in [74]. conceptual dependencies (CD) is a technique used to represent content or meaning. The technique lends itself well to physical descriptions and actions, but does not include means to describe high-level concepts such as goals. A fixed structure is used to describe an event, and multiple structures are combined to describe more complex events.

Conceptual graphs are a flexible method of knowledge representation using bipartite graphs developed by John F. Sowa and based on a combination of the existential graphs of Charles Sanders Peirce and semantic networks [75]. One class of vertexes represent concepts and the other represents relationships between these concepts. A conceptual graph (CG) has labelled relation nodes, labelled concept nodes, and numbered arcs that link relations to concepts. Please refer to Chapter 4 for further details.

2.5 Human Feedback

The use of feedback information or interaction with the user allows the expression of meaning to be shared between both the hand gestures and the feedback information provided. This allows a simple set of gestures to express meaning beyond what each combination of gesture sequences can provide.

To eliminate the need for people to learn a new gesture language and to simplify the gestures required in a human machine interface, visual feedback can be utilized [76]. The machine interpreting the gestures, a television, identifies an open hand and provides a visual icon representing the hand and options that can be selected by moving the hand. Although the method proposed in [76] simplifies the interface, the hand is reduced to a simple pointer, similar in function to a joystick or mouse. An interface of this type is suitable in simple, highly constrained human machine interfaces such as controlling the volume and channel on the television, but is not easily scalable to more complex interactions and does not provide a significant advantage over a wireless or strategically placed joystick or mouse.

2.6 Outstanding Issues In Human-Robot Interaction

Limitations in existing approaches hinder the ability of a device to understand human expression. Addressing these limitations will reduce the time we spend and effort we exert to communicate intent to devices in our inevitable ubiquitous computing environments.

2.6.1 Reliability

Current research inadequately handles information conveyed across multiple channels using multiple modalities. This inadequacy results in reliability limitations that must be addressed. The popular method of independent recognition of each modality or channel followed by keyword substitution [58] discards relevant information that can be used in mutual disambiguation to improve reliability.

2.6.2 Human Feedback

Most human-machine interface research concentrates primarily on the transfer of information from human to machine. Although this is a vital component in natural communication with machines, research seldom considers the system as a whole. In systems that consider human feedback, input interfaces are not natural and rely primarily on conventional tactile input [77]. The incorporation of effective human feedback as part of the input interface is essential to ensure communication with machines feels natural. In face-to-face communication, we receive continual feedback in the form of verbal, facial expressions, and other body language cues. Communication is a closed-loop feedback system, and all parts of this system must be considered to realize natural communication.

2.6.3 Segmentation

Most existing approaches perform temporal segmentation of each separate input modality or channel as a distinct step before meaning is determined. Use of n-gram stochastic models have been used to assist in some research [42], but this is only useful with popular sequences of symbols (the sparse data problem) or in languages with rigid syntax. This limits the information available to segment channels to the syntax of recognized information and does not take advantage of complementary or redundant information encoded in other modalities or the overall meaning. Only very specific forms of human expression can be described using rigid syntax based rules.

2.6.4 Concurrency

Alternative modalities are currently treated either in a similar manner to speech or as supplemental information tied directly to a primary communication channel. Hand gestures are normally considered to be a sequence of logical units, one after another with each unit exclusively occupying a region of time. Even many hierarchical approaches, where observation sequences are classified then passed to a second stage classifier, do not explicitly address concurrency in human hand gestures. Facial expression approaches that classify expressions into one of the six basic prototypical emotions may also limit temporal flexibility. These common approaches make simplifications to prevent logical units from co-existing, and ignore the inherent concurrency in expressions. Although this concurrency limitation may not conflict with the interpretation of some whole word signs in a formal sign language, it is not a reasonable limitation to impose on the interpretation of meaning from natural hand gestures and mixed facial expressions.

2.6.5 Scalability

Representation of each hand gesture may perform well with a small set of gestures, but this approach does not scale well. When facial expressions are also considered, the increased complexity further reduces scalability. A low-complexity approach is required to allow an understanding system to scale to practical sizes. Addressing scalability concerns will allow machines to interpret a wider range of expressions and intentions. As an example, ASL uses approximately 6000 distinct gestures to represent single words, but on-line approaches usually deal with 100 or fewer gestures [39, 42]. Use of PaHMM techniques may address some of the limitations in scalability by representing signs in ASL using *movement* and *hold* sections for each hand, but is limited to using the hand as an atomic unit [45].

2.6.6 Understanding

Considerable effort has been invested in hand gesture recognition by direct application of pattern recognition techniques, but without leveraging the underlying meaning [78]. Similar attention to the importance of understanding the underlying meaning is also emphasized in speech and written language research. High-level meaning or knowledge information must be applied to the low-level recognition data to resolve ambiguities and produce a meaningful interpretation. A significant limitation in current natural human-machine communication research is the emphasis on low-level recognition and omission of this higher-level information to understand the meaning of human expression. In research that considers high-level information, it is handled as a sequential step that strictly follows low-level recognition. Unfortunately, there is no clear natural division between low-level recognition and high-level understanding. Imposing an arbitrary sequential division, as is common in current research, limits the information that can be leveraged to understand human expression.

Chapter Three

Perceiving Human Expression

The process of sensing and perceiving is essential to enable a robot to interact within its environment, including the environment's human inhabitants, which are essential for human-robot interaction. Before delving into issues specific to perceiving human expression, it is important to understand how and where perception fits into the overall approach of this research. After describing perception in the context of the overall architecture, this chapter focuses on the flow of information from sensing through to the extraction of relevant spatio-temporal features and concludes with the recognition of concurrent elements of human expression.

3.1 Approach

The ability of a robot to understand natural communication is limited both by the information it collects from human expression and by the effectiveness of synthesizing and analyzing this information. Appropriate integration of information expressed across diverse natural communication methods is critical to ensure a robot can robustly understand human expression. The author's research advances the state-of-the art in natural human-robot communication by combining potentially conflicting information available across multiple human modes and modalities at a conceptual level in a knowledge-based closed-loop framework. To address reliability issues, a knowledge-based closed-loop architecture and supporting algorithms are applied to explicitly handle the inherent concurrency and conflicts that exist in natural communication. Redundant and complementary information contained in facial expressions and hand gestures are combined with domain specific knowledge and refined with feedback. Although basic facial expressions and hand gestures are used in experiments, the focus of this research is on the overall architecture for combining information both between multiple sources and between observations and a priori information.

To maintain flexibility, this multimodal integration architecture ensures modality specific logic is loosely-coupled to downstream components. This approach ensures the addition or substitution of alternate sources of human information remains as simple as possible, including future incorporation of speech, tone of voice, body language, gaze, or brain emission sources. The approach presented in this thesis is inspired from human face-to-face communication, where humans attempt to understand intent by selecting relevant information, organizing it, and integrating that information with appropriate existing knowledge [79].

The ability to robustly understand the intent of natural human communication will benefit almost every device with which we interact. One of the more practical and flexible devices that can benefit from natural communication is the personal robotic assistant. A personal robotic assistant is a robot that interacts with people to perform a wide range of tasks. Since it may be complicated to explain your intent to a personal robotic assistant using traditional interfaces, this is an ideal device to demonstrate the benefits of multimodal fusion toward understanding natural human communication.

Seamless integration of personal robotic assistants into human society relies on effective and transparent human-robot communication [77]. Communication using natural multimodal methods will reduce human effort and time required to communicate with a robot when compared against conventional single and multimodal interfaces. Improving the robustness of the multimodal human-robot interface increases the tolerance of vague or ambiguous communication. Similarly, improving flexibility permits information to be incorporated from a diverse range of sources in practical environments. A reduction in human effort and improvement in personal service robot transparency is realized with the presented design of a flexible and robust architecture to fuse multimodal information for the purpose of understanding human intent.

Ongoing research toward improving the transparency of human-robot communication include use of voice based interfaces [80], deictic hand gestures [81], hand gesture languages, facial expression recognition, and some multimodal approaches [63, 67, 82, 83]. Major differences between current human-robot communication research and the approach presented in this thesis include:

1. Explicit use and quantification of uncertainty and vagueness in knowledge representation and reasoning as applied to information encoded across multiple modalities,

2. Support for flexible co-ordination and integration of input information at different stages in the understanding process, and

3. Use of iterative feedback to explicitly handle conflicts between both multiple channels and between observations and current context.

3.2 Scope

Although parallels are often drawn between human face-to-face communication and human-machine communication, the purpose and environment in which these these two forms of communication occur must be considered. Human face-to-face communication is normally carried out in a co-operative environment, where nearby people are silent and do not interfere with the communication. However, we often desire to communicate with machines in environments or at times where human face-to-face communication is either not possible or very difficult. Imagine a room full of people, all trying to communicate with their personal service robots using only voice. Without additional sources of information, it would be very difficult to determine the content and intent of any individual. Understanding human intent expressed across multiple modalities provides the flexibility necessary to overcome limitations of uni-modal approaches to support robust communication in real-world environments.

The scope of this research includes the design and development of a knowledge-based architecture to flexibly integrate information about natural human communication. Natural hand gestures and facial expressions are used in experiments to evaluate the performance of this architecture. These two sources of information are selected to emphasize the importance of non-verbal human expressions that are often overlooked as secondary or lesser sources of human expression [84]. The scope also includes the design and implementation of a closed-loop iterative feedback mechanism, in addition to adapting an existing knowledge representation and reasoning method to help understand the intent of human communication. Existing low level recognition algorithms for hand gestures and facial expressions are adapted to maintain the focus of this research.

3.2.1 Iterative Feedback

The distinction between syntax, semantics, and pragmatics of natural languages is seldom clear even when dealing with a single modality. However, current research frequently handles the analysis of natural language in distinct and isolated steps. Incorporating iterative feedback into the understanding process helps to reduce discrepancies between expressed and understood intent. Hypotheses of intent are used to iteratively refine low-level recognition and synthesis tasks, including segmentation and disambiguation. These same hypotheses are used to resolve conflicting information expressed from different modes in addition to helping co-ordinate and fuse information between multiple sources.

3.2.2 Flexible Integration

Rather than fixing one location to integrate information from multiple sources, the framework explored in this research can support integration of multimodal information at three possible levels: at the feature level, between recognized primitives, and at the conceptual level. This approach introduces flexibility to balance the computational complexity of fusion against the quality of the resulting information. Considering the need to bring together information from more than just hand gestures and facial expressions ensures the framework does not impose unnecessary restrictions when extending sources of information to include speech, lip movements, brain emissions, and other forms of human expression.

3.2.3 Conflict Detection and Resolution

Conflicts are inevitable in human communication. The presented approach deliberately identifies and explicitly incorporates conflicts as an important component of the architecture. Rather than attempting to ignore conflicts, they are leveraged to improve the quality of the resulting device derived intent. Conflicts can be distinguished at three different levels: between primitives in different channels (mode conflict), between concepts and surrounding context (temporal conflict), and between the resulting action performed and the original human intent (action conflict). Mode and temporal conflicts are used to provide feedback and refine the understanding of human expression.

3.2.4 Knowledge Representation

Using appropriate knowledge representation and reasoning techniques assists understanding by using both complementary and redundant information from multiple modalities. A common multimodal approach is to rely on a single primary modality for most information except for specific predefined cues [58–60]. When specific cues are detected, information from a second modality is often substituted. Our approach combines information from multiple modalities not solely for substitution of information, but to achieve a more accurate result by resolving ambiguities through the combination of available information and prior knowledge. Although this approach is applied to personal robotic assistants, both the approach itself and its results are useful to bring us one step closer to useful ubiquitous computing environments.

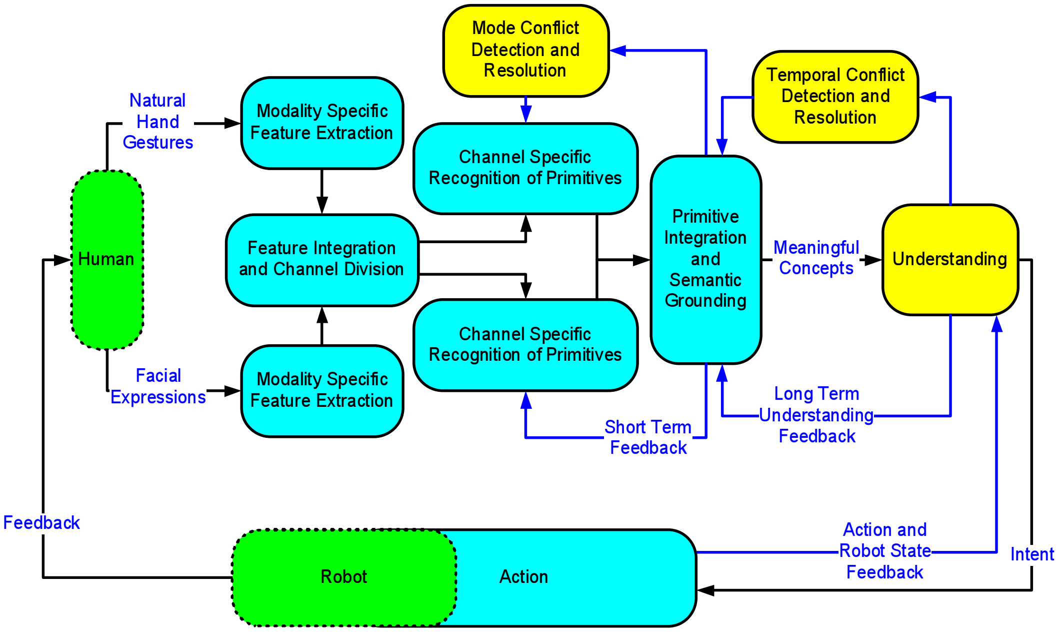

Mode Conflict Detection and Resolution
Temporal Conflict Detection and Resolution
Natural Hand Gestures
Modality Specific Feature Extraction
Channel Specific Recognition of Primitives
Human
Feature Integration and Channel Division
Primitive Integration and Semantic Grounding
Meaningful Concepts
Understanding
Channel Specific Recognition of Primitives
Facial Expressions
Modality Specific Feature Extraction
Long Term Understanding Feedback
Short Term Feedback
Feedback
Action and Robot State Feedback
Intent
Robot
Action

Figure 3.1: The high-level architecture consists of multiple modality specific feature extraction modules that provide information over multiple channels to flexible recognition, semantic grounding, and understanding components. Iterative feedback is used to address conflicts, to refine the understanding of human expression, and to guide the integration process.

3.3 Overview of Architecture

A high-level overview of the architecture used in this research is illustrated in Figure 3.1. Spatio-temporal information is obtained from visual sources of facial expression and glove-based sources of hand movement and configuration. Specific features are identified from these concurrent streams of information using modality specific algorithms and provided to a channel divider. The channel divider can be used in future research to optimize the balance between the number of concurrent features in each channel, and the number of channels. Primitives are recognized and brought together to form concepts. The recognition process uses feedback based on detected conflicts and relationships between primitives and concepts. Integration is possible at different stages in the process to help balance the complexity of combining multimodal information against the potential loss of information during recognition and synthesis. Integration during understanding and reasoning reduces the dependence on specific modalities, but is not always more robust than integration closer to the raw signal where more information is available. The modality independence of this architecture is emphasized in the design to easily incorporate alternate and additional sources of human information.

Knowledge representation and reasoning algorithms are adapted to explicitly handle uncertainties across multiple sources of information. Knowledge to support semantic grounding and understanding of intent is represented in a domain specific ontology along with relevant logic in the form of conceptual graphs and fuzzy logic. Fuzzy logic is used rather than strict logical assertions to allow the unavoidable uncertainty in the world around us to be explicitly modelled. This information about the uncertainty of expert knowledge is useful to evaluate and select a reasonable interpretation of human expression from viable alternatives.

This architecture is partially motivated from the growth point theory [6] which suggests that all forms of human expression originate from a single *growth point*, or point at which the underlying intent starts to form. Since the growth point theory focuses on a thought or intent, it is independent from a specific form of expression or even language itself. The theory also suggests that the actual thought or intent does not necessarily develop in the same order as expressions are conveyed. This thesis considers multimodal human expressions as observable outcomes that originated from a single intent or growth point. The order of the expressions and concepts is not critical during the search to determine the underlying intent that created the observed human expression.

At a high level, the author's approach consists of several components loosely-coupled to facial expressions and natural hand gestures. These components include signal processing and feature extraction, recognition of primitives, semantic grounding, conflict analysis, and understanding of intent. Each of these components are brought together in a multi-level integration system to make use of combined data when appropriate. Clearly defined and simple interfaces between each of these components ensures our architecture can be extended and easily modified by incorporating focused advancements in any of these specific areas. Future research providing incremental improvements to any one of these components can be easily incorporated to further improve the overall performance of the multimodal system for understanding human expression.

3.3.1 Real-World Applications

Implementation and experimental evaluation of theoretical work and algorithms takes advantage of both tethered and vision based human data acquisition methods, and is applied in a domestic service robot or personal robotic assistant scenario. Existing robot control algorithms, and high level task planning approaches can be integrated into the architecture illustrated in Figure 3.1. Redundant and complementary information is obtained from multiple natural human communication modes and modalities, including hand gesture, and simple components of facial expression.

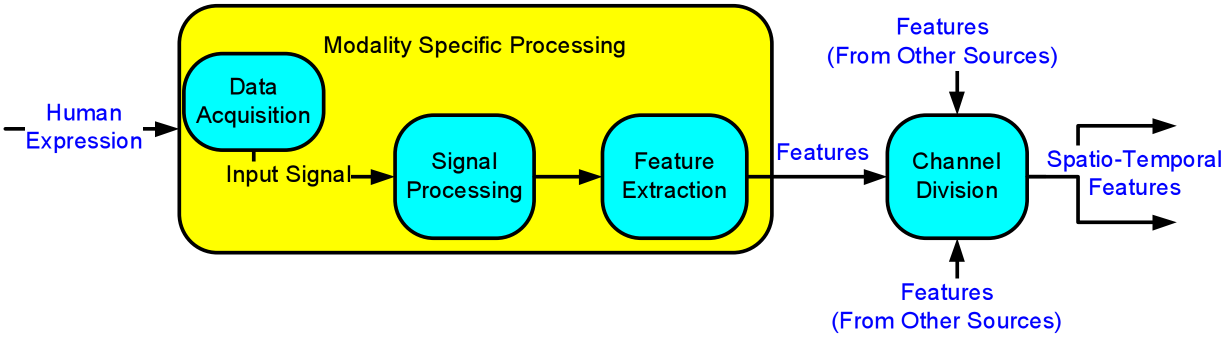


Figure 3.2: The sensing and preprocessing activities specific to each modality are encapsulated in separate modality specific modules. Every modality specific module outputs a sequence of features over time.

This architecture allows the robot to carry out desired actions based on understanding human intent expressed across multiple modalities.

Practical examples of communication scenarios include communication between a person in the rapidly increasing elderly demographic and a personal service robot. An elderly person might require assistance with meal preparation, pet care, general cleaning, garbage removal, control of more complex devices, and delivery of specific objects including water and medication. Similar scenarios exist both with disabled people and people that might wish to offload specific tasks to a personal service robot. Since the direction of this research is headed toward transparent and natural human-robot communication, a wide range of communicated intent is possible, limited only by the knowledge and capabilities of the personal service robot.

Key aspects of communicated intent might be noisy, occluded, or not originally expressed using a single modality. Information in a single modality often requires qualification. Complementary information to help qualify or disambiguate this information may be available from deictic gesture, lip movement, speech, sign language, or other form of human expression. Performance of the presented approach is evaluated by communicating intent using a combination of hand gestures and facial expressions. Robustness to incomplete and conflicting information is evaluated using occluded and omitted information and controlled additive noise.

3.4 Sensing and Conditioning Individual Modalities

Information expressed in each modality is observed and processed in a modality specific manner to identify useful features. These features are subsequently used to recognize primitives and synthesize meaningful concepts in separate modules. This section describes the modality specific logic used to extract useful features from human hand gestures and facial expressions. An overview of these components is shown in Figure 3.2, starting from the sensing of human expression and input signal processing, and resulting in a set of spatio-temporal feature vectors for use in recognition.

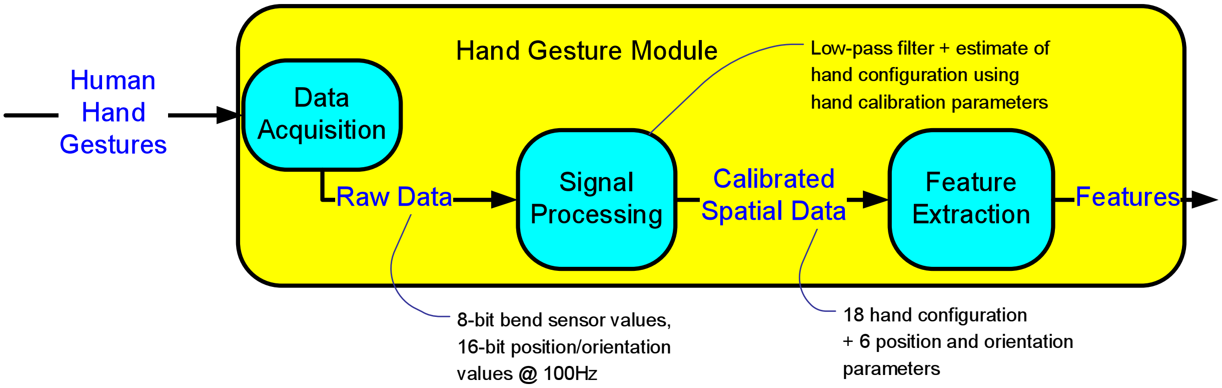

Human Hand Gestures
Hand Gesture Module
Data Acquisition
Raw Data
Signal Processing
Calibrated Spatial Data
Feature Extraction
Features
Low-pass filter + estimate of hand configuration using hand calibration parameters
8-bit bend sensor values, 16-bit position/orientation values @ 100Hz
18 hand configuration + 6 position and orientation parameters

Figure 3.3: Human hand gestures are translated to useful features by acquiring joint angle information, smoothing, and transforming to relevant frames of reference.

3.4.1 Processing Hand Gestures

Since the generation of hand gestures is driven by human thought processes, it is reasonable to hypothesize that hand gesture information can be analyzed to infer meaningful groups of concepts. Unfortunately, deriving these concepts directly from human hand gestures is a nontrivial task. The author's previous research in hand gestures [85] is enhanced to leverage dependencies between observations and to use a more flexible set of concurrent primitives, and extended to introduce flexible reasoning and take advantage of feedback from the integration, conflict analysis, and understanding aspects of this multimodal framework.

The primary input to the hand gesture specific processing module is a continuous stream of natural hand gestures, and the primary output is a set of features arranged in concurrent channels. Channels are currently defined based on physical sources of features and concurrency in existing human annotation systems, including a channel for arm dynamics, and a separate channel for index finger dynamics. It is possible to extend this approach in future research to dynamically adjust the mapping between features and channels based on feedback from recognition and understanding. This feedback can also be used to incorporate both the integration of information from other sources and the state or action of a target physical device.

As illustrated in Figure 3.3, the feature extraction component measures hand gesture properties and produces a sequence of calibrated spatial data. Encapsulating feature extraction in this manner isolates all higher level tasks from the physical data acquisition method. This isolation allows the presented approach to focus on inference of intent and multimodal issues independent from its physical implementation.

Although it is desirable to use an unobtrusive interface in a real-world application, the field of hand gesture recognition using vision-based interfaces requires considerable advancement to overcome outstanding occlusion and accuracy obstacles. The author developed a glove based feature extraction platform, illustrated in Figure 3.5 to provide a sequence of calibrated spatial data. This spatio-temporal data includes critical hand and wrist joint angles coupled with orientation and position of the hand relative to known reference points. As vision based hand gesture feature extraction systems mature, this glove-based approach can be seamlessly replaced with a vision based 3D-reconstruction approach [30–32, 86].

3.4.1.1 Acquiring Hand Gesture Observations

An 18-sensor CyberGlove and a Flock of Birds[1] as shown in Figure 3.5 is used to obtain accurate 3D representations of the hand position and configuration. Joint angles are provided by the CyberGlove to the author's software as a set of 8-bit values. Two stages of calibration are applied, one at the hardware level before the values are exposed, and one in the data acquisition software module. Hardware calibration specifies the offset and voltage range to use during A/D conversions, and the software calibration is used to convert the quantized 8-bit values to joint angles in radians. A low-pass filter is used to help reduce the impact of high frequency noise. As shown in Equation 3.1, the angle in radians $\theta_{i,t}$ at time t for each joint angle $i = 1 \cdots 18$ is determined from the magnitude of the resistive bend sensor $d_{i,t}$ using an offset b_i and a scaling factor α_i, as shown in Equation 3.1.

$$\theta_{i,t} = \alpha_i(d_{i,t} - b_i) \tag{3.1}$$

The calibration parameters α_i and b_i depend on the hand being measured, but a set of sample values are included in Table 3.1. The offsets b_i and scaling parameters α_i are determined from a specific hand by first sampling the range of raw values $d_{i,t}$ available during the full range of motion in each joint. Limits on the range of motion in each joint, including angles at the extreme endpoints are used to calculate initial values of b_i and α_i. These initial values are manually refined as needed using a computer generated 3D model of the hand until contact between the thumb and each real fingertip is accurately represented in the 3D model. This simple calibration process helps accommodate differences in hand size that affect the placement of each resistive bend sensor relative to the joints they measure. Each of the measurements referenced in Table 3.1 are illustrated in Figure 3.4. The distal interphalangeal (DIP) joint is not measured, but approximated as a function of the proximal interphalangeal (PIP) and metacarpophalangeal (MCP) joints using the empirically derived relationship shown in Equation 3.2 [87].

$$\theta_{DIP} = 0.3\theta_{PIP} \tag{3.2}$$

Orientations are obtained from the Flock of Birds using a quaternion representation to reduce the effect of orientation on precision. Position information is provided by the Flock of Birds using one 16-bit value for each of the X, Y, and Z axes. An affine transformation is applied to use a body-centric frame of reference at the subject's shoulder. To simplify this transformation, experiments are conducted with the subject in a known position and orientation relative to the Flock of Birds transmitter. Position values are translated to real world units by linearly scaling the 16-bit values and compensating for any fixed disturbances. A radius of 914mm is used to support

[1] CyberGlove is a registered trademark of Immersion Corporation (`http://www.immersion.com`). Flock of Birds is a registered trademark of Ascension Technology Corporation (`http://www.ascension-tech.com`)

Table 3.1: Values for the user-dependent calibration parameters for each joint angle are determined by comparing measured raw values against known hand structure limitations with visual verification.

Joint (i)	Description	α_i	b_i
0	Thumb CMC (basal) flexion (radial abduction)	0.0159	21.0
1	Thumb MCP joint flexion	0.0145	96.9
2	Thumb IP joint flexion	0.0143	109.0
3	Thumb-Index CMC (palmar) abduction	0.00505	81.0
4	Index MCP joint flexion	0.0162	70.0
5	Index PIP joint flexion	0.0146	77.0
8	Middle MCP joint flexion	0.0173	63.0
9	Middle PIP joint flexion	0.0180	80.0
11	Middle-Index MCP joint abduction	0.00450	151.0
12	Ring MCP joint flexion	0.0176	75.0
13	Ring PIP joint flexion	0.0179	91.0
15	Ring-Middle MCP joint abduction	0.00616	162.0
16	Pinkie MCP joint flexion	0.0166	85.0
17	Pinkie PIP joint flexion	0.0182	80.0
19	Pinkie-Ring MCP joint abduction	0.00600	131.0
20	Palm Arch	0.00912	71.0
21	Wrist Pitch	0.00869	143.0
22	Wrist Yaw	0.00394	89.0

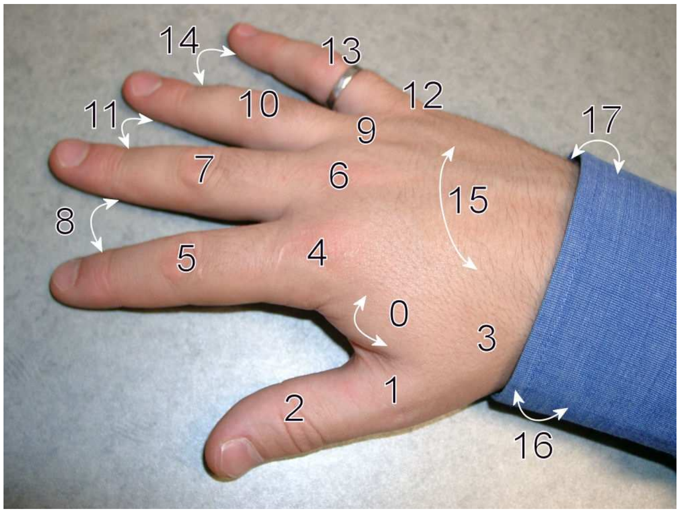


Figure 3.4: The 18 joint angles measured by the glove hardware include all but the four distal interphalangeal joints.

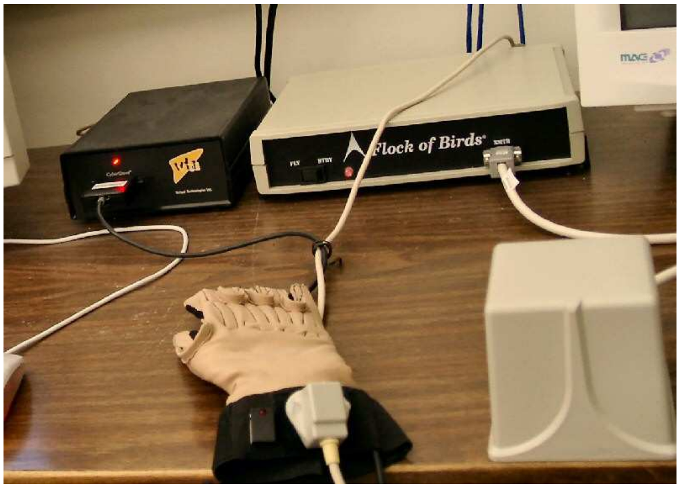


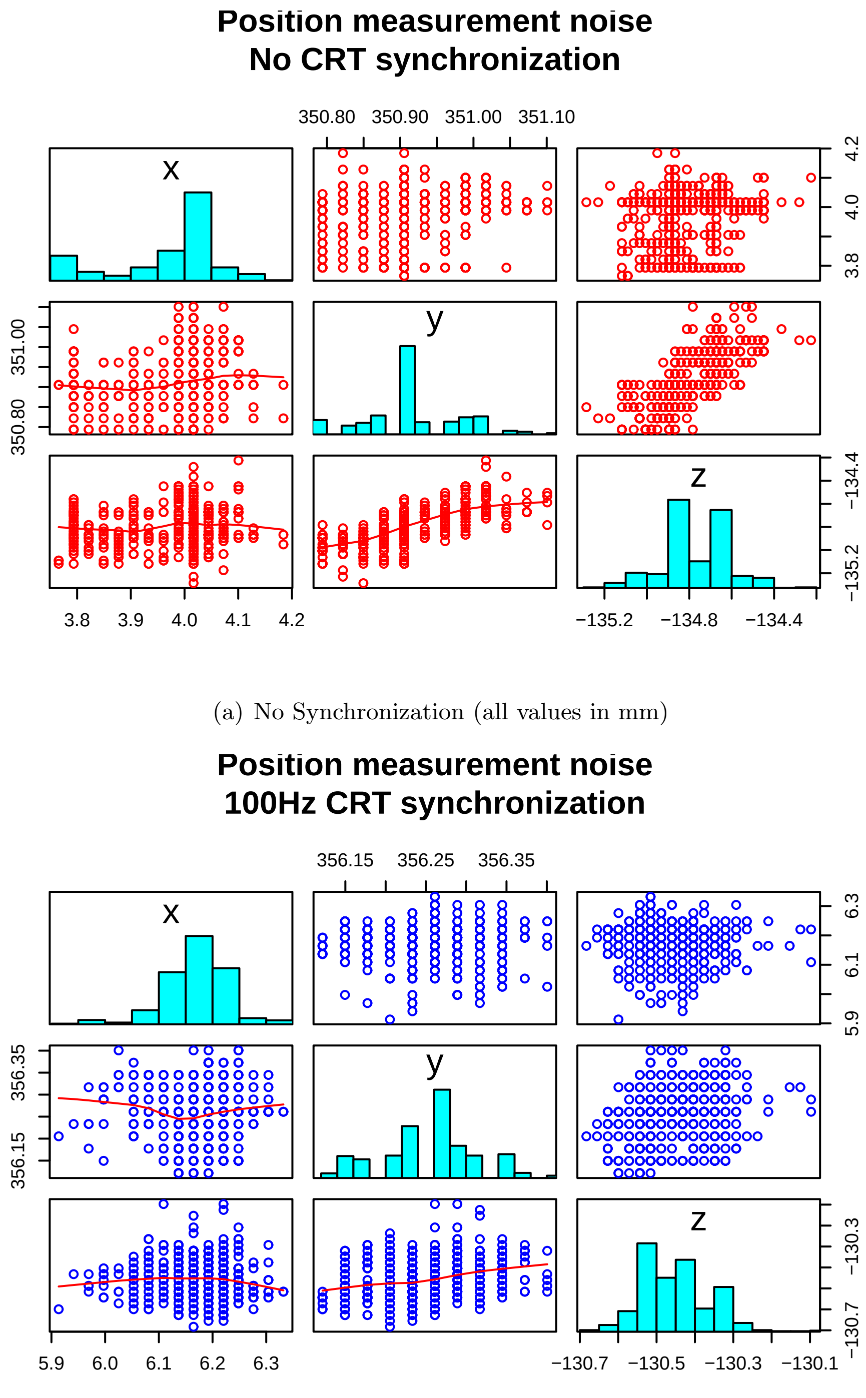
Figure 3.5: The Flock of Birds uses the DC pulsed electromagnetic source in the lower right corner together with the small directional sensor attached to the base of the glove to simultaneously determine 3D position and orientation. The CyberGlove uses strategically placed resistive bend sensors to measure hand and wrist joint angles. Hardware controllers for both devices are shown at the top, communicating with the computer over dedicated serial connections.

full arm extension in any direction without torso movement. A larger radius is supported by the hardware, but cannot be justified against the increased quantization error and susceptibility to noise.

Electromagnetic interference from the closest CRT is minimized by timing the measurement of hand position and orientation to avoid the electromagnetic disturbance caused during each vertical refresh. The effects of the monitor disturbance are shown in Figure 3.6(a). This figure illustrates the precision in the measured position of the sensor in mm when the sensor position does not change relative to the emitter. After synchronizing the measurements with the closest CRT, the noise has a lower correlation between co-ordinates. The synchronization process resulted in no significant change in precision, but a systematic change in absolute position as shown in Figure 3.6(b). The level of noise present in the feasible hand movement space is significantly smaller than the variances in repeated dynamic or static human hand gestures. However, systematic errors in the absolute measurement are mitigated by measuring positions and angles relative to the subject's shoulder.

3.4.1.2 Defining Useful Hand Gesture Features

Low-level features are calculated directly from the position, orientation, and configuration of the hand. The first and second derivatives of each measurement are estimated and included as low level features to provide some short-term dynamics to help recognize primitives. The flexion or extension of each of the 4 fingers is extracted using the angle of the MCP and the PIP joints. The distal joint on each finger is not included as a feature as it is highly dependent on the MCP and PIP joints. This dependency can be easily tested by trying to move the fingertip portion of



(a) No Synchronization (all values in mm)

(b) Synchronized at 100Hz (all values in mm)

Figure 3.6: Stationary hand position measurements are subject to normally distributed white noise after synchronizing measurement timing against the closest CRT at 100Hz.

your own finger without bending the other joints or vice-versa.

The abduction and adduction, or the angle between fingers, is included as low-level feature to provide a measure of the spread of fingers. Thumb features include one degree of freedom for the interphalangeal and MCP joint angles, and two degrees of freedom for the trapeziometacarpal (CMC) joint. Two degrees of freedom in wrist movement are also included, along with a measure of palm curvature to determine the position of each joint with respect to the forearm. The position relative to the forearm is required since the position and orientation measurement of the hand is obtained at the end of the forearm before conversion to a body centric frame of reference. A summary of the derived features used in the author's framework and their relationship to the acquired hand gesture data is provided in the following list of descriptions, with examples illustrated in Figure 3.7. These derived features can be directly calculated from observations, unlike the static and dynamic primitives that are recognized within sequences of features.

Finger Closure The angular deviation of a line formed between the tip of a finger and its MCP, scaled between the maximum closure, where the fingertip is in contact with the inside of the palm, and minimum closure, where the MCP joint is hyper-extended, and the PIP joint is extended as far as possible.

Finger Curvature Distance between a fingertip and the MCP joint. Finger curvature is distinct from finger closure, as a finger can have maximum curvature but minimum closure when the MCP joint is hyper-extended.

Fingertip-Thumb Proximity Distance between a fingertip and the endpoint of the thumb. Finger-thumb proximity is an important feature in many hand gestures, including the dynamic hand gestures for chicken and eat.

Fingertip The position of of the endpoint of the distal segment of a finger in spherical coordinates relative to the shoulder $(r_{f,i}, \phi_{f,i}, \theta_{f,i})$. This information is valuable when combined with finger orientation to help resolve deictic hand gestures.

Palm Roll The orientation of the palm, measured as an angle of rotation around a line formed by the extension of the forearm.

Palm Orientation The spherical angles between the normal vector of the face of the palm and the shoulder frame of reference, indicating the orientation of the palm in space.

Palm Direction The spherical angles between the vector between the base of the palm and the MCP of the index finger and the shoulder frame of reference, indicating the direction of the palm.

Shoulder Proximity The position of the centre of the palm relative to the shoulder in spherical coordinates. Spherical coordinates have been selected rather than Cartesian coordinates since the spherical approach provides information more relevant to shoulder and elbow joint configurations.

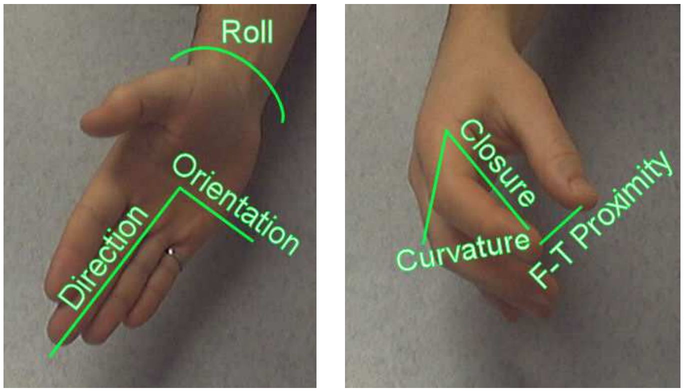


Figure 3.7: Features are derived from hand gesture data to simplify recognition. These derived features include hand configuration details such as finger curvature in addition to whole hand movement details like the position of the hand with respect to the shoulder.

Subsets of the directly observed and derived hand gesture features are used to recognize static and dynamic hand gesture primitives in the next module of the framework. Before moving on to a description of the recognition module, the second modality specific module will be presented to extract useful features from human facial expressions. Additional modality specific modules can be incorporated into this architecture to sense and perceive the remaining forms of human expression.

3.4.2 Processing Facial Expressions

In natural communication between two people, a number of complex expressions are expressed not only by the hands, but also by the appearance of the face. These facial expressions convey valuable information about human emotional state and intent [77]. Understanding facial expressions helps to infer the underlying intent, complementing information provided through other sources of human expression. As illustrated in Figure 3.8, the external interfaces and components of the facial expression processing module are designed similar to the previously described hand gesture processing module. Visual facial information is processed to arrive at a set of concurrent features.

As shown in Figure 3.8, the feature extraction component used in this research encapsulates all hardware-specific functionality to support seamless interchange with alternative feature extraction approaches. Facial expressions are acquired using a Sony DVI-30 camera, a standard visible-spectrum camera that provides a full 24-bit per pixel video signal to avoid artifacts associated with lossy compression sources, and to avoid loss of chromatic information as seen with YUV420 formats. Use of this standard hardware provides sufficient detail to obtain major visible facial features with the exception of precise eye gaze.

Existing face detection, segmentation and tracking algorithms are implemented to provide calibrated spatial data for recognition. Since our approach is emphasizing the understanding and integration of information between multiple modalities, a frontal upright face pose is assumed. This simplifying assumption can be eliminated in the future without modification to the structure of higher level modules required to understand human expression.

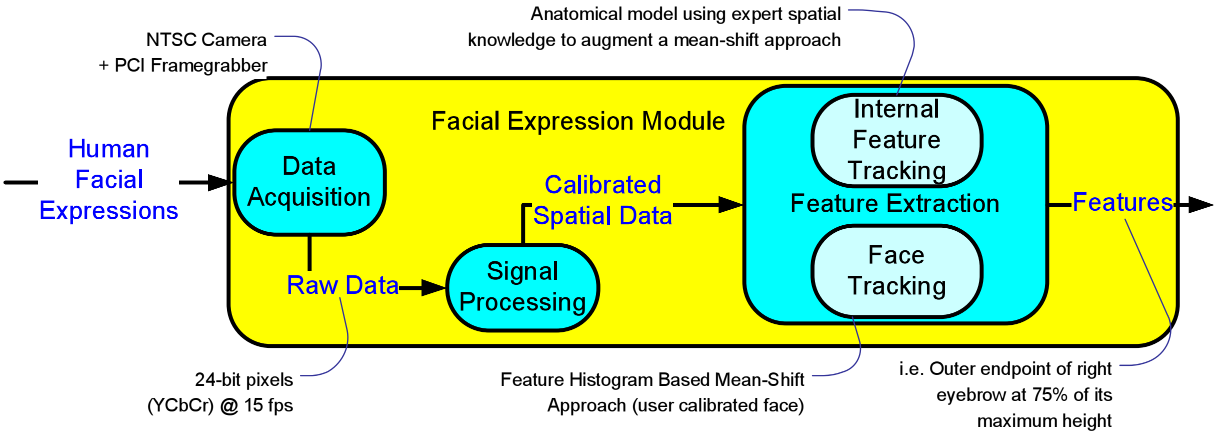

Anatomical model using expert spatial knowledge to augment a mean-shift approach
NTSC Camera + PCI Framegrabber
Facial Expression Module
Human Facial Expressions
Data Acquisition
Calibrated Spatial Data
Internal Feature Tracking
Feature Extraction
Face Tracking
Features
Raw Data
Signal Processing
24-bit pixels (YCbCr) @ 15 fps
Feature Histogram Based Mean-Shift Approach (user calibrated face)
i.e. Outer endpoint of right eyebrow at 75% of its maximum height

Figure 3.8: Facial expressions are translated into useful features by tracking the face in images and measuring relationships between key locations within the face.

Three steps are performed to simplify the data intensive incoming video stream into shape and location changes of key features on the face. The first of these steps locates and tracks the face as a whole (when key features inside the face become unreliable or lost). The second locates and tracks key points on the eyebrows, eyes, nose, and lips, and the final step calculates relationships between these key points to provide shape and location changes of key features to the recognition module.

3.4.2.1 Detecting, Isolating, and Tracking the Face

Initially, the face must be located without prior knowledge of its location in the image. A robust and computationally simple approach to detection and tracking is the use of infrared to detect pupils [21]. Unfortunately, the infrared pupil detection and tracking approach requires specialized equipment. Automatic detection and tracking of pupils and other key facial features is more challenging when restricted to a conventional visible spectrum camera.

A feature histogram based mean-shift approach as presented in [88] is used to detect and re-initialize individual facial feature trackers when required. This approach is used rather than the popular Viola-Jones face detection algorithm [89] or particle based methods [90] due to the low complexity, minimal training data requirements, and robustness of the feature histogram based mean-shift approach to camera motion. As shown in [88], the feature histogram based mean-shift approach to detection and tracking can be implemented with a linear average computational cost in the number of target pixels in the face and number of scale variations. Although this approach is susceptible to loosing the face when occluded, it is practical for many forms of multimodal human-robot communication, and can be extended in the future to take advantage of additional face motion information, including use of Kalman (traditional, extended or unscented) or particle filter techniques.

This approach is based on comparing the probability distribution of features in a known face model against the distribution of features in candidate faces in the image, and transforming the candidate face to maximize the similarity. Probability distributions are represented using histograms of desired properties of the known face ($\vec{q}$) and candidate face ($\vec{p}$) areas. Since the histograms (with m total bins) are estimations of probability distribution functions, they must sum to 1.

The known face model is manually defined by identifying an elliptical area with height h_y and width h_x around a known face (approximately). This area is then normalized to a unit circle and weighted to place emphasis on the centre area and less emphasis on ambiguous areas at the perimeter that may be affected by background noise or occlusion. This weighting process is performed by applying an isotropic kernel to the image. As recommended in [88], a kernel with an Epanechnikov profile ($k_E(x)$, convex and monotonically decreasing - see Equation 3.3) is used.

$$k_E(\vec{x}) = \begin{cases} \frac{1}{2c_d}(d+2)(1-x) & x \leq 1 \\ 0 & otherwise \end{cases} \tag{3.3}$$

The resulting Epanechnikov kernel is shown in Equation 3.4, using $d = 2$ (2D image application) and $c_d = \pi$ (the volume of a unit circle).

$$K_E(\vec{x}) = \begin{cases} \frac{2}{\pi}(1-|\vec{x}|^2) & |\vec{x}| \leq 1 \\ 0 & otherwise \end{cases} \tag{3.4}$$

The histogram with a predetermined number of bins (m) is then built using Equation 3.5 [88]. q_j is the probability of a feature represented by histogram bin j. $\vec{x_i^*}$ is the normalized vector of the location of the i^{th} pixel in the target model of n pixels. $quantize(\vec{x_i^*})$ is the quantization function used to assign a bin to the features given by the i^{th} pixel, and δ is the Kronecker delta function.

$$\begin{aligned} q_j &= \frac{1}{\sum_{i=1}^{n} k(|\vec{x_i^*}|^2)} \sum_{i=1}^{n} k(|\vec{x_i^*}|^2)\delta\left(quantize(\vec{x_i^*}) - j\right) \\ q_j &= C \sum_{i=1}^{n} k(|\vec{x_i^*}|^2)\delta\left(quantize(\vec{x_i^*}) - j\right) \end{aligned} \tag{3.5}$$

The histogram is built to represent a possible face in the image with centre $\vec{x_0}$ using Equation 3.6 [88]. This equation differs from Equation 3.5 only with the introduction of a scaling parameter (h) and offset ($\vec{x_0}$). $\vec{x_i}$ represents the normalized vector of the position of the i^{th} pixel in the possible face area of n_h pixels. The pixel locations $\vec{x_i}$ are normalized within a unit circle using the target model normalization parameters (h_x, h_y).

$$\begin{aligned} p_j(\vec{x_0}) &= \frac{1}{\sum_{i=1}^{n_h} k(\left|\frac{\vec{x_0}-\vec{x_i}}{h}\right|^2)} \sum_{i=1}^{n_h} k(\left|\frac{\vec{x_0}-\vec{x_i}}{h}\right|^2)\delta\left(quantize(\vec{x_i}) - j\right) \\ p_j(\vec{x_0}) &= C_h \sum_{i=1}^{n_h} k(\left|\frac{\vec{x_0}-\vec{x_i}}{h}\right|^2)\delta\left(quantize(\vec{x_i}) - j\right) \end{aligned} \tag{3.6}$$

The goal of face detection, segmentation, and tracking is now reduced to finding the location ($\vec{x_0}$) and scale (h) that maximizes the similarity between the target histogram ($\vec{q}$) and the histogram of the possible face area ($\vec{p}$). A histogram similarity metric based on the Bhattacharyya coefficient [88, 91] is used to compare histograms[2]. This metric, as shown in Equation 3.8 provides a distance measure ($d(\vec{p}(\vec{x_0}), \vec{q})$) between the histograms of each possible face location $\vec{x_0}$ and the target face histogram $\vec{q}$ [88].

$$d(\vec{p}(\vec{x_0}), \vec{q}) = \sqrt{1 - \rho(\vec{p}(\vec{x_0}), \vec{q})} \tag{3.7}$$

Substituting $\rho(\vec{p}(\vec{x_0}), \vec{q}) = \sum_{j=1}^{m} \sqrt{p_j(\vec{x_0}) q_j}$ results in:

$$d(\vec{p}(\vec{x_0}), \vec{q}) = \sqrt{1 - \sum_{j=1}^{m} \sqrt{p_j(\vec{x_0}) q_j}} \tag{3.8}$$

The smooth nature of the kernel (Equation 3.4) and similarity measure (Equation 3.8) allows simple and computationally efficient gradient based optimization methods to adjust x_0 and maximize similarity. $\rho(\vec{p}(\vec{x_0}), \vec{q})$ from Equation 3.7 should be maximized to maximize this similarity.

As shown in [88], a Taylor expansion of the Battacharyya coefficient around the last known face location ($\vec{x}_{last}$)can be used under the assumption that the face does not move far between observations. This linear approximation is shown in Equation 3.9, and simplified to Equation 3.10 by ensuring $p_j(\vec{x}_{last}) > 0$ and substituting Equation 3.6.

$$\rho(\vec{p}(\vec{x_0}), \vec{q}) \approx \frac{1}{2} \sum_{j=1}^{m} \sqrt{p_j(\vec{x}_{last}) q_j} + \frac{1}{2} \sum_{j=1}^{m} p_j(\vec{x_0}) \sqrt{\frac{q_j}{p_j(\vec{x}_{last})}} \tag{3.9}$$

$$\rho(\vec{p}(\vec{x_0}), \vec{q}) \approx \frac{1}{2} \sum_{j=1}^{m} \sqrt{p_j(\vec{x}_{last}) q_j} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left(\left| \frac{\vec{x_0} - \vec{x_i}}{h} \right|^2 \right) \tag{3.10}$$

Where,

$$w_i = \sum_{j=1}^{m} p_j(\vec{x_0}) \sqrt{\frac{q_j}{p_j(\vec{x}_{last})}} \delta \left(quantize(\vec{x_i} - j) \right).$$

The mean shift method applied in [88] is used to iteratively calculate $\vec{x}_{next}$ using only the portion of Equation 3.10 dependent on $\vec{x}_{last}$ as shown in Equation 3.11. To compensate for gradual changes to the appearance of the face over time, the continuously adaptive mean shift extensions are incorporated into the implementation [92].

$$\vec{x}_{next} = \frac{\sum_{i=1} n_h \vec{x_i} w_i}{\sum_{i=1} n_h w_i} \tag{3.11}$$

The stopping criteria for the mean shift iteration consists of both a maximum number of iterations and a minimum change between $\vec{x}_{next}$ and $\vec{x}_{last}$. The features used in the histogram representation of the face include the chromatic components (C_b and C_r) of the face area in a YC_bC_r colour space. Texture information can be incorporated if needed in future research without modifying the underlying approach. It should be noted that inclusion of texture information substantially increases computational complexity.

[2] $d(x, y)$ is a metric since it satisfies the three conditions: 1. $d(x, y) \geq 0$ and $d(x, y) = 0 \iff x = y$, 2. $d(x, y) = d(y, x)$, and 3. $d(x, y) + d(y, z) \geq d(x, z)$

Table 3.2: Key points tracked for internal facial components

Facial Component	Key Points
Lips	Left endpoint
	Right endpoint
	Centre of upper lip
	Centre of lower lip
Eyebrows	Left endpoint
	Right endpoint
	Mid-left peak
	Mid-right peak
	Mean of inner endpoints
Eyes	Centre of left eye
	Centre of right eye

3.4.2.2 Localizing Internal Facial Features

Representation of facial features can be broken into three general categories: anatomical or physical models based on key points [29], 3D reconstruction [93, 94], and whole face representations [28]. Whole face representations require extensive face databases for training and result in models that are not easily interpreted. 3D models can be very accurate, but are not suitable for an online implementation due to their computational complexity. Anatomical models based on key points may not offer the same level of accuracy as 3D models, but result in human interpretable models with low computational complexity and minimal training data requirements.

Rather than approaching facial expressions from a whole-face perspective, separate features within the face are identified and tracked independently within constraints based on human face configurations. This individual component approach allows information about visible facial features to be used even when other features are partially occluded. Each facial feature is identified with a set of key points as is frequently performed in face analysis research [95, 96]. These key points, as enumerated in Table 3.2 are sufficient to determine basic facial expressions. To keep the computational cost of feature extraction low and take advantage of information calculated during face detection, segmentation, and tracking, a similar histogram based mean-shift algorithm is used to detect and track the key points identified in Table 3.2.

Multidimensional receptive field histograms are applied to facial features and used to represent each key point within its local appearance context [14]. Receptive fields are built using six local operators: The chromatic components (C_b and C_r) of the area in a YC_bC_r colour space, and the first derivative of the Gaussian operator (Equation 3.12) in the $\theta = 0$ and $\theta = \frac{\pi}{2}$ directions relative to the given area at two different scales ($\sigma = \sigma, 2\sigma$) as shown in Equation 3.13. This set of local operators is chosen to provide information about local feature structure, including a measure of colour, edge intensity and orientation. The two different scales provide relationships between high and low frequency intensity gradients, whereas the two orientations provide sufficient information to describe the direction of each intensity gradient.

$$G^{\sigma}(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.12}$$

$$G^{\sigma}_{\theta}(x,y) = \frac{\partial}{\partial v} G^{\sigma}(x,y) \tag{3.13}$$

Expert knowledge constrains the initial search and recovery of lost key points, including restricting nose tracking to just below the middle of the face and restricting lip tracking to the lower half of the face. Additional constraints of face configuration as described in [95] are also applied to improve initial detection and recovery of tracked points on each key facial feature. Sub-pixel precision is achieved by setting the distance based stopping criterion for the mean shift iteration (Equation 3.11) to less than the distance between adjacent pixels. The output of this feature extraction module contains a parametric representation of each facial feature using the spatial location and velocity of key points listed in Table 3.2.

3.4.3 Moving Beyond Modality Dependent Processing

As described in the extraction of hand gesture and facial expression features, each modality specific module handles the necessary signal preprocessing and extracts features that can be used to recognize primitives for higher level analysis. The facial expression features are extracted based on geometric properties of key points on the face (an anatomical approach). These features ensure intermediate results remain human interpretable. Similarly, hand gesture features are based on human interpretable joint angles, finger properties, and hand position relative to the forearm and shoulder. The concurrent features provided from each of these modules are selected to remain as independent as possible from the physical sensor.

Abstracting features from the specific sensor allows the underlying data acquisition components to be replaced with alternatives that may be more suitable in specific applications. For example, a glove based hand gesture system is impractical for brief human-robot interaction or for applications that require full hand mobility. In these scenarios, the data acquisition component can be replaced with a vision-based 3D reconstruction approach without redefining the features available for the subsequent recognition module. Similarly, the underlying components of the facial expression module can be exchanged with a vision source using a different spectrum, or augmented with an electromagnetic or other head tracking system. The resulting features from the modality specific modules are not based on the sensor, but on characteristics of the observed human modality. Most modifications to the underlying data acquisition techniques will change the accuracy and precision, but do not require fundamental changes to the feature definitions.

3.5 Identifying Primitives In Feature Streams

When a sequence of calibrated low-level features is available from one or more modality specific processing modules, it is analyzed to identify previously defined patterns. Rather than attempting to identify complete, meaningful concepts directly from low-level features, the author divides this problem into two stages; recognition of basic, concurrent elements followed by a semantic

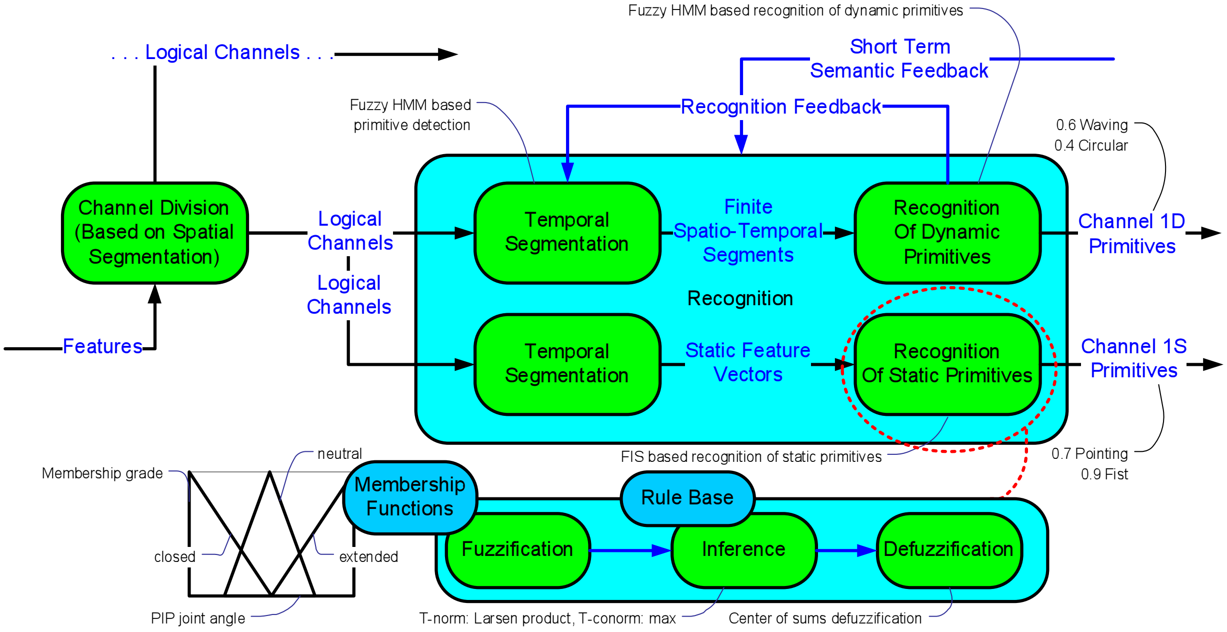


Figure 3.9: Each stream of features is analyzed to identify possible primitives. Dynamic primitives are identified in features spanning short durations of time, and static primitives are identified in features at a specific instant.

grounding process that combines these elements to form meaningful concepts. The author intentionally selected basic patterns to ensure the total number of patterns to identify is maintained at a feasible level. In almost any classification problem, performance will not improve with an increase in the number of classes as it becomes more difficult to discriminate between classes. The bounded set of basic patterns or primitives are used as building blocks to form meaningful concepts in subsequent processing.

The hand gesture and facial expression recognition modules recognize and output sets of concurrent primitives when given a time sequence of calibrated human expression features. Each primitive is associated with a specific time instant or interval and a possibility of existence. These primitives may overlap in time or exist together to support the natural concurrency in both hand gestures and facial expressions.

Several tasks are involved in the successful recognition of primitives, as shown in Figure 3.9. Recognition is based on logical channels rather than specific modalities to support direct recognition of patterns based on more than one modality, and recognition of patterns in a subset of features available in a single modality. Use of these channels helps to maintain flexibility and to keep each modality as loosely coupled to recognition as possible. In the scope of this research, each logical channel is defined as a subset of the features available from a single modality specific processing module.

Two different classes of primitives are recognized: static and dynamic. Static primitives describe a component of human expression at one instant in time, whereas dynamic primitives describe a component of human expression conveying information over a time interval. Although the same set of features can be used as input for recognition, these two classes of primitives have very different characteristics, and should be handled using appropriate techniques. Static primitives, such as an extended finger, or a left-facing palm can be conveniently described using linguistic terms with precise meaning but vague relationships to the underlying features. A rule-based fuzzy inference system is used to take advantage of human knowledge in the form of vague linguistic terms to precisely define and associate a measure or possibility to each static primitive.

Dynamic primitives, on the other hand, are not as easily described in linguistic terms. The temporal aspect of dynamic primitives must be considered, however, variations occur in both the value of each feature and its timing. The HMM technique is currently used to recognize patterns of varying lengths in other forms of human expression, including speech, handwriting, and even analysis of deoxyribonucleic acid (DNA). The ability of a HMM to robustly accommodate variations in sequence lengths is valuable for the recognition of dynamic primitives. The forward generative models allow the set of dynamic primitives to be modified without requiring extensive retraining of the complete set as is often required with an artificial neural network approach. Unfortunately, the underlying independence assumption between observations of a HMM is difficult to justify when considering human facial expressions and hand gestures. To relax this independence assumption, the author applies and evaluates a fuzzy adaptation of HMMs to train and recognize dynamic primitives in human expressions. In addition to relaxing the independence assumption, this fuzzy adaptation has been shown to train faster than a traditional HMM in other domains [97, 98].

3.5.1 Defining Useful Primitives

The desired output of language primitives recognized from hand gesture channels are defined based on expertise in the human hand gesture annotation community. Most annotation systems are designed for a specific sign language or other form of communication, but some exist that describe basic hand gestures in a language neutral manner. One of the more comprehensive and language independent annotation systems, the Hamburg Notation System [99] (HamNoSys) includes static and dynamic annotations at a level of abstraction suitable for the recognition module. This notation system explicitly identifies concurrency in hand gestures, and provides a good trade off between complex symbols, and human interpretability. This thesis adapts many of the symbols in the HamNoSys originally intended for human annotation to define hand gesture primitives for automatic recognition by a robot or other interactive device.

Several systems exist to transcribe, or annotate hand gestures in writing, including HamNoSys, Stokoe [100], and SignWriting [101]. Stokoe is specific to the ASL (although some variants exist for other languages), whereas SignWriting and HamNoSys attempt to be as language independent as possible. The HamNoSys was originally developed as a research tool, and, unlike SignWriting, it is not designed for everyday human-human communication. Its design focuses on the hand motions and configurations in a language independent manner. One application of this annotation system is to specify hand gesture and other human expression information to control a virtual avatar [102]. Although the problem of human expression generation may appear very different than the problem of recognizing human expressions, both problems share similar representation or annotation issues. The HamNoSys provides the necessary flexibility to describe static and dynamic hand gestures without imposing unnecessary constraints on the language itself. As demonstrated with its prior suitability for hand gesture generation in virtual avatars, the annotations system provides sufficient support to accurately describe generic hand gestures.

Both static and dynamic facial expression primitives are defined using the channels of key geometric feature points provided from the feature extraction module. Static primitives, including eyebrows raised, or lips extended are recognized directly from knowledge in fuzzy rules applied to the currently extracted facial feature information. Dynamic primitives recognized from facial expression channels, including contracting lips or raising and lowering of the eyebrows during a yawn, are recognized using a fuzzy hidden Markov model (FHMM) technique.

Although the facial expression primitives described here may share some similarities with FACS action units, they are designed at a higher level of abstraction. Action units are based on specific muscle movements that can be challenging to unobtrusively measure, recognize, and relate to a specific facial expression. The identified primitives in this research provide adequate information for very simple facial expressions, but will need to be revisited if discrimination between a more comprehensive set of facial expressions is desired.

3.5.2 Modelling Dynamic Primitives

The sequences of features provided after processing each modality are inherently non-stationary signals. HMM techniques are well suited for modelling non-stationary signals such as these feature sequences, as variations in signal properties (observation values) over time can be handled by dynamically adjusting individual sub-sequences and modelling the distributions of the observation values. This robustness to minor temporal variations is a major strength of a HMM approach, and required when considering any form of human expression. No two dynamic hand gestures are expressed with exactly the same timing, and neither are multiple dynamic facial expressions. A rigid time bound does not appropriately describe all instances of opening the mouth, nor is it applicable to constraining the length of all arcing hand motions toward the face. Dynamic primitives must be correctly recognized in hand gestures and facial expressions even when produced at varying speeds.

A HMM can be considered a finite state machine in which the state is not directly observable, but indirectly inferred through another stochastic process [103]. Each state in a HMM represents one component of the underlying process which, in the context of human expression, is the underlying human translation of intent to measurable process. When considering hand gestures and facial expressions, the measurable process is generated by muscle contractions in the arm, hand, and face.

In a HMM, each model (λ) is described using three parameters,

$$\lambda = (\mathcal{A}, \mathcal{B}, \pi) \tag{3.14}$$

Where $\mathcal{A} = \{a_{ij}\}$ is the state transition probability matrix, $\mathcal{B}$ the matrix of probabilities of generating each observation symbol, and π the initial probabilities of being in each state [103].

The state transition probability matrix (Equation 3.15) defines the probability of s_j being the next state at time $t+1$ (q_{t+1}) given that the current state at time t (q_t) is s_i,

$$a_{ij} = P(q_{t+1} = s_j | q_t = s_i), i, j \in \{1..N\} \tag{3.15}$$

Similarly, the emission matrix $B = \{b_{jk}\}$ (Equation 3.16) defines the probability of generating each observation v_k from a given state s_j.

$$b_{jk} = P(v_k | q_t = s_j), i \in 1..N, k \in \{1..M\} \tag{3.16}$$

The three fundamental tasks in a HMM are: to determine model parameters that maximize the likelihood of a set of training observations when given the model, to determine the likelihood of a model when given a sequence of observations, and to find the most likely sequence of states within a model when given a sequence of observations. In the recognition task, the probability of a given observation sequence $\mathcal{O}$ is calculated for each model $\lambda_i = (\mathcal{A}, \mathcal{B}, \pi)$ to determine the model with the highest likelihood $P(\mathcal{O}|\lambda_i)$. The training task can be tackled using an iterative procedure to optimize the values in the transition matrix $\mathcal{A}$, emission matrix $\mathcal{B}$, and initial states π.

One of the challenges to modelling a process using a HMM is not the three fundamental tasks, but selecting an appropriate topology. Two commonly used topologies are the ergodic and left-right topologies. A left-right topology describes a process that progresses from one state to the next without reversing or returning to previous states. Since most facial expressions and hand gestures progress from some initial state forward to some final state, using the left-right topology is reasonable for the dynamic primitives being modeled. The ergodic topology, on the other hand, is fully connected and can be used to model more complex processes. If sufficient training information is available, a fully connected topology can be applied in all models. Unfortunately, the number of parameters that must be learned in a fully connected topology is exponential in the number of states and impractical for most training sets. Nevertheless, a small ergodic topology is used in the noise models to capture hand gesture and facial motions that are not currently of interest. Each dynamic primitive is modeled using a left-right topology, as shown in Figure 3.10. Although opportunities exist for optimization of this topology to more accurately represent individual dynamic primitives, the left-right topology is used successfully in a wide range of speech and hand gesture recognition applications [85, 104, 105].

One of the major assumptions of a traditional HMM is that each observation is independent, and observations are independent from prior states. The author applies a variation of the traditional HMM technique to recognize dynamic human expression primitives by estimating the likelihood of each primitive over time. The variation uses fuzzy measures and integrals to relax the independence assumption, since it would be difficult to justify in respect to natural human expression. This generalization of the HMM was originally introduced to tackle the problem of recognizing continuous handwriting [97, 98]. These generalized hidden Markov models are able to take advantage of the complex dependencies that exist between states in dynamic human expression primitives. For simplicity of implementation, the Choquet integral is selected as the

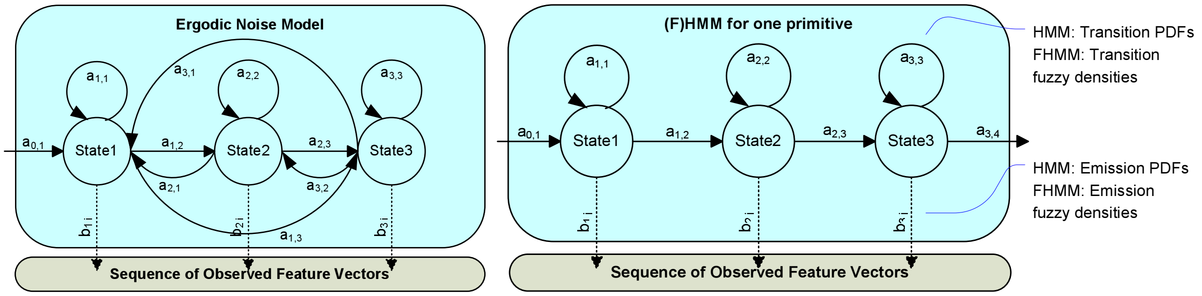


Figure 3.10: Each dynamic primitive, i, is modeled using a fuzzy hidden Markov model λ_i with a left-right topology. Ergodic topologies are used to model noise.

fuzzy integral, the possibility measure as the fuzzy measure, and Larsen product as the fuzzy intersection operator to recognize dynamic human expression primitives.

The extracted features are partitioned into logical channels based on their relevance to known hand gesture and facial expression primitives. Although the complete set of features can be used as observations to every model, the partitioning is performed to reduce the size of the observation vector and the associated problem of training a large set of parameters. The set of features used to model an upward arc of the hand does not include finger curvature, and the model for forming a fist does not need the distance from the shoulder to the centre of the palm. Each of these channels contains features used as a sequence of observation vectors for training or analysis using each known FHMM. Each primitive (i) is modeled by a single FHMM (λ_i), where each state of the model is not directly observable and represents one sequential component of the possible dynamic primitive. The observations ($\mathcal{O}$) correspond to values of the previously defined features at regular time intervals.

The key points being tracked on the face are represented as sequences of observation vectors for use in the FHMM framework. Similarly, hand configuration, orientation, and (body-centric) position information is also represented as sequences of observation vectors $(O_1, O_2, \cdots, O_T)$. Each observation vector represents a small interval of time (67ms for facial expression channels, 10ms for hand gesture channels), with O_T representing the most recent observed features, and O_1 the oldest. These vectors encode angles and trajectories between relevant tracked key points for the face, and joint angles, positions, and orientations for the hand as previously described in the defined features. Absolute distances and positions are seldom meaningful for either facial expressions or hand gestures, as they vary considerably across individuals, scale, and orientation. The one notable exception to absolute position information is deictic hand gestures, where the absolute trajectory of a finger can refer to a physical entity. Although absolute measurements are not used to recognize primitives, this information is readily available for use beyond the scope of this research, including the important tasks of anchoring and planning.

Observations can be represented in either discrete or continuous form. A discrete observation is modeled using a finite set of known values, whereas a continuous observation is modeled using parametric distributions of values. A continuous value does not have to be modeled using a continuous HMM, and can be easily quantized for use in a discrete model. Unfortunately, the quantization process requires knowledge of how features will be used in each model to generate a good codebook or process for quantization. This codebook must be revised to accommodate changes or additions of models.

Since the values encoded in each feature are continuous in nature, including the motion of the hand, lips, or eyebrows, it is logical to model these forms of human expression using a continuous representation. Use of a continuous representation allows additional dynamic primitives to be incorporated into the system and existing ones to be removed without retraining previously trained primitives. A continuous representation allows the observations to be used directly, without additional quantization error and without maintaining a global codebook. Although the goal of this module is to identify the best primitives in each channel, each primitive is still modeled independently when using a continuous representation. Training a new dynamic primitive does not require information about or interaction with any of the alternate primitives.

Separate models are used for each possible primitive in each channel. For facial expression channels, a separate model is used for the transitions of the mouth, transitions of the eyebrows, and one model is used to capture background noise. Similarly, separate models are used to represent each dynamic hand gesture primitive. Each of these models (λ) are initially trained using pre-labelled observations. At any point in time, the probability of each primitive is calculated as the conditional probability of the current observation sequence given each traditional model $P(\mathcal{O}|\lambda)$. Similarly, the possibility of a transition toward each expression for a single FHMM is calculated using Equation 3.17,

$$\hat{P}(\mathcal{O}|\hat{\lambda}) = \sum_{i=1} \hat{\alpha}_t(i)\hat{\beta}_t(i) \tag{3.17}$$

As derived in [97], the fuzzy forward ($\hat{\alpha}$) and backward ($\hat{\beta}$) variables used to classify human expression primitives are shown in Equation 3.18 and Equation 3.19, respectively.

$$\hat{\alpha}_{t+1}(j) = \left(\sum_{i=1}^{N} \hat{a}_{ij}\rho_t(i,j)\hat{\alpha}_t(i) \right) \hat{b}_j(O_{t+1}) \tag{3.18}$$

$$\hat{\beta}_t(i) = \sum_{j=1}^{N} \hat{a}_{ij}\rho_t(i,j)\hat{\beta}_{t+1}(j)\hat{b}_j(O_{t+1}) \tag{3.19}$$

The function ρ is what differentiates this fuzzy adaptation of HMMs from traditional HMMs. ρ is a non-linear function of $\hat{\alpha}$ and $\hat{a}$, describing the relationship between a fuzzy measure defined on the set of observations $(O_1, O_2, O_3, \ldots, O_t)$ and a fuzzy measure defined on the set of states $(S_1, S_2, S_3, \ldots, S_N)$. It is this function that allows the statistical independence assumption of the traditional HMM to be relaxed [97]. In the author's implementation, $\rho_t(i,j)$ is defined using the Choquet integral with respect to the possibility measure. The Larsen product is used as the intersection operator.

3.5.3 Modelling Static Primitives

Static primitives such as an extended finger, the hold position of deictic gestures, or a fist configuration do not depend on time. Since dynamic time warping is not required to align sequences, the complexity of a HMM is unnecessary. Since static primitives can be described well using linguistic terms and, in fact, many are already described in linguistic terms in human annotation tasks, it is advantageous to select a technique that can leverage these linguistic terms. Static primitives are modeled using a fuzzy rule-based inference system. In most fuzzy rule-based inference systems, inputs are fuzzified and a set of conditions are evaluated to derive new facts. These new facts are then aggregated, and defuzzified if a crisp output is desired. To recognize static primitives, the conditions, or antecedents are based on the current set of features and the consequents describe static primitives. Mamdani inferencing is used with the Larsen product implication operator along with centre of sums defuzzification to take into account overlapping areas from different rules. The centre of sums defuzzification method is shown in Equation 3.20, where $\mu_{C'_k}(z_i)$ is the consequent membership function for the k^{th} rule at point z_i in the universe of discourse.

$$z^*_{(COS)} = \frac{\sum_{i=1}^{n} z_i \cdot \sum_{k=1}^{n} \mu_{C'_k}(z_i)}{\sum_{i=1}^{n} \sum_{k=1}^{n} \mu_{C'_k}(z_i)} \tag{3.20}$$

Example rules include:

- *If the finger closure for all 4 fingers is medium, and the finger curvature for each finger is high, then the cupped fingers primitive is high*,
- *If the fingertip-thumb proximity is touching for each of 4 fingers, and the curvature is low, and the closure is medium, then the pinching grasp is high*,
- *If the lip left-right distance is large, then the lips extended primitive is high*, and
- *If the left eyebrow-lip proximity is low and the right eyebrow-lip proximity is low, then the eyebrows lowered primitive is high.*

Since it is difficult to precisely define membership functions, the original membership functions based solely on human expert knowledge can be refined in a supervised training process to further improve performance without loosing the benefits of a human readable model.

3.5.4 Training Models and Recognizing Primitives

With almost any form of human expression, ambiguity exists not only in the content of the expression but also in its timing. To eliminate subjective manually labelled observations, facial expressions and hand gestures are collected and tagged automatically. Computer generated visual prompts are used to request randomly selected primitives. Facial expression and hand gesture information is automatically tagged in the observation sequences based on the visual prompt to simplify validation and testing and reduce the need to manually annotate observation sequences.

The stochastic nature of HMM initialization and training can influence the comparison between HMM variants. To reduce training process discrepancies, each HMM variant is trained to the same convergence threshold using identical training sets. Training performance can be measured using the total iterations required to converge, and when optimizations are identical in both implementations and total CPU time required over many repetitions. A jackknife procedure can be used to evaluate recognition performance on each of the test sets when excluded from training. Recognition performance can be measured using the accuracy of the most likely dynamic primitive, or the top N most likely dynamic primitives over a given time frame for a known expression.

3.5.5 Using Primitives After Perception

Concurrent primitives are recognized in each input channel using a combination of fuzzy hidden Markov models for dynamic primitives and a fuzzy inference system for static primitives. A separate model is maintained for each primitive in each previously defined channel. Although different recognition approaches can be used for each incoming channel, the combination of the FHMM and fuzzy inference system (FIS) techniques performs well for channels derived from either hand gesture or facial expression sources. Since primitives are explicitly concurrent, there is no need at this point to directly compare the outcomes produced from different channels or different algorithms.

Human hand gestures can efficiently convey complex concepts, excelling with both direct spatial references, symbolic representations, and abstract notions. Primitives are loosely based on the notation system HamNoSys to ensure each primitive has some meaning to a human observer. Primitives are recognized in facial expressions by applying the same dynamic and static recognition algorithms to geometric features of the face. These primitives are used in subsequent modules to arrive at semantically grounded concepts, and ultimately, an estimate of the intent of the human participant.

Although it is possible for a person to express their intent using hand gestures alone, or even just facial expressions, it is not as natural or intuitive as using these or other multiple forms of human expression simultaneously. It is very difficult to determine intent solely from observing facial expressions or from hand gestures alone. A stern look may be expressed because a serious command is being issued, or it may be expressed because the person is getting mad about their intent being misinterpreted by a robot. This modality specific processing module identifies plausible primitives, but additional information is required to resolve conflicts between primitives and ambiguities about concepts and the underlying intent. In almost all formal sign languages, information is conveyed using hand gestures in combination with other forms of body language. In common face-to-face interactions, information is conveyed using speech in combination with facial expression and body movements. Both complementary and redundant information exists between primitives derived from facial expressions and primitives derived from hand gestures. Effective integration between multiple sources of human expression is essential to help maximize the accuracy and robustness of inferred intent.

Chapter Four

Understanding Intent

> Great things are not done by impulse, but by a series of small things brought together
>
> — Vincent Van Gogh

The co-operative act of communicating with a robot involves exposing information about human intention through natural expression. The preceding chapter followed the flow of information from perception of these human expressions through to the recognition of basic elements or primitives. Continuing toward understanding these human expressions, this chapter focuses on the conceptual-level aspects of the overall approach. The set of small concurrent primitives are accepted from the perception modules, and brought together to form hypotheses of intent as shown in Figure 4.1. Cleanly abstracted from physical interfaces and specific modalities, the process to form concepts from primitives is presented and discussed. This chapter continues to follow the flow of information exposed through human expression, leading to the author's approach to apply human knowledge with precise meaning but vague values to help infer hypotheses of desired human intent.

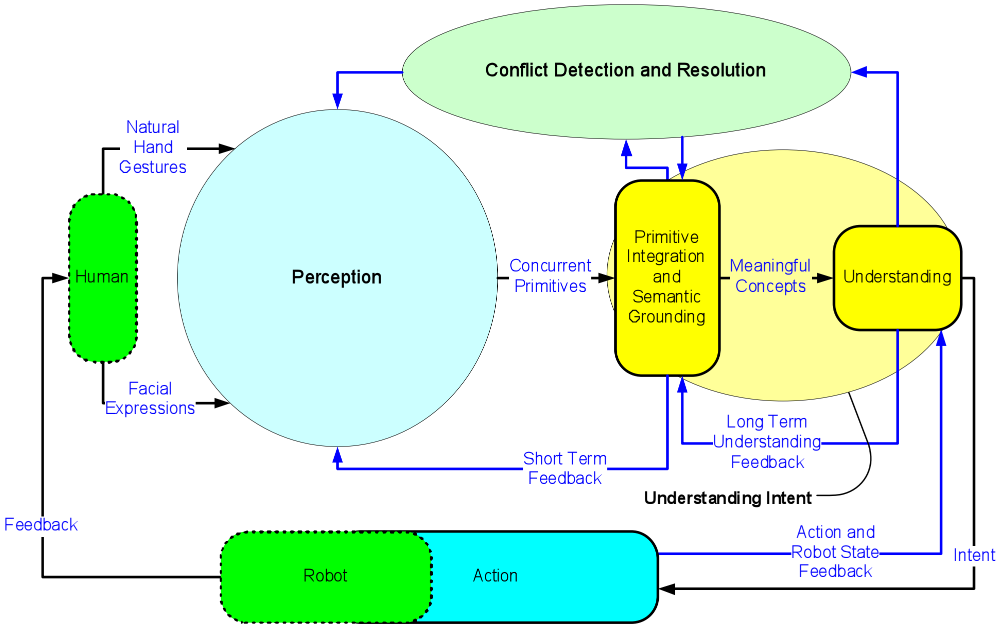


Figure 4.1: The modules responsible for understanding intent rely on inputs of concurrent primitives from the perception modules to form hypotheses of intent based on domain specific knowledge.

4.1 Forming Concepts By Integrating Primitives

> **fusion,** *n.*
> a merging of diverse, distinct, or separate elements into a unified whole
> **integrate,** *v.*
> to form, coordinate, or blend into a functioning or unified while
> — Merriam-Webster Online Dictionary

Information is distributed across multiple modalities during the expression of human ideas. Although it is possible to analyze each of these modalities independently to infer the intended content of the communication, this naïve approach may not be optimal. Potentially useful information contained in the correlations, or lack thereof, between modalities may be ignored. The accuracy and the level of certainty of the robot's understanding of expressed concepts can benefit from analyzing and incorporating information contained in both relationships between modalities and information available in the form of domain specific knowledge.

Current approaches to multimodal user interfaces attempt to make use of a variety of different input modalities. Speech and deictic gesture is most common, but gaze, facial expression, and emotion are also occasionally explored. Each modality can provide either new, complementary, or redundant information about the concepts being expressed. In the redundant case, integration can reduce uncertainty and improve accuracy. In the complementary case, integration can help disambiguate noisy or underspecified content. Multimodal integration can improve the overall robustness of the human-robot interface.

Psycholinguistic research provides interesting insights into multimodal integration. Multimodal integration, when approached from a psycholinguistic perspective, concentrates on determining a common source from which expressions in each modality are generated. Current research emphasizes the catchment feature model as a tool for multimodal fusion at the discourse planning and conceptualization level [64]. Since the catchment feature model can be used to combine information close to the source of generation, a significant portion of the approach deals with determining common sources of generation.

The more common approach to the integration of human expression information is to impose a syntactical model of human communication, and rely on this model to combine multiple sequential concepts. This approach has its roots in the analysis of unimodal expressions, including speech, handwriting, or facial expressions on their own. Unfortunately, building an accurate model of the syntax of multimodal communication requires a prohibitive volume of sample data, and is simply not feasible for all forms of human expression, including spontaneous hand gestures and facial expressions.

Rather than restricting integration to a single location, or to a sequential analysis, the author's approach can support integration of information at three different levels: when forming concepts from primitives, when forming analogies from concepts, and when forming primitives from observed features. This approach allows loosely related information, such as facial expressions and hand gestures to be evaluated at a high level, while tightly-coupled sources of information, such as lip movements and speech to be evaluated at a low level. In this thesis, the author focuses on

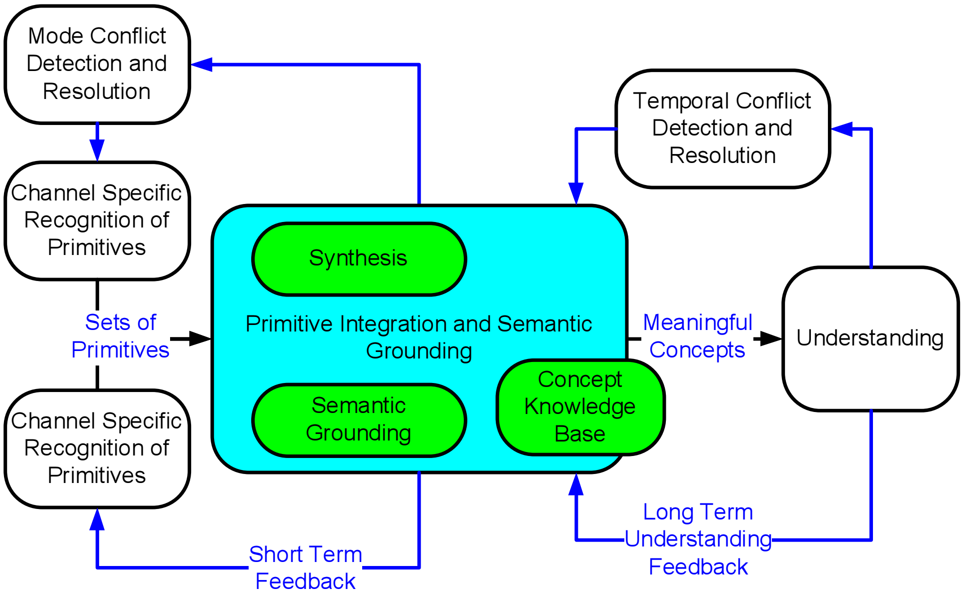

Mode Conflict Detection and Resolution
Channel Specific Recognition of Primitives
Sets of Primitives
Channel Specific Recognition of Primitives
Synthesis
Primitive Integration and Semantic Grounding
Semantic Grounding
Concept Knowledge Base
Temporal Conflict Detection and Resolution
Meaningful Concepts
Understanding
Short Term Feedback
Long Term Understanding Feedback

Figure 4.2: Multimodal integration combines sets of concurrent primitives to form potential concepts. These concepts are provided to the understanding module to help determine the meaning of the observed human expressions.

the two higher levels of integration; integration of concurrent primitives to form concepts, and the integration of concepts to form analogies.

When bringing primitives together to form concepts, not only is the identified primitive important, but also is the time interval in which it was detected and a measure of certainty in the presence of the primitive. Uncertainty is not only present in the observations, but it is inevitable in the definitions of concepts in terms of human expressions. A fuzzy representation is used to explicitly model this uncertainty in relationships between primitives and the concepts they define. Using this representation helps leverage existing expert knowledge and minimize requirements for sample data. Relationships between high level concepts and most forms of natural expression are difficult to accurately model using a parametric probability distribution. The fuzzy representation approach allows the integration process to take advantage of a human-generated knowledge base with the ability to encode expert knowledge within the representation. In addition, use of a fuzzy representation with expert knowledge provides information to support rapid training with a relatively small data set. Use of probabilistic and belief approaches often rely on very large data sets that may not always provide adequate coverage, or may require prohibitive time and resources to generate.

The remainder of this section concentrates on the integration of primitives to form concepts as shown in Figure 4.2. The following section uses these concepts to form hypotheses about the underlying meaning. This flexible approach can be applied to support integration at both low and high levels. The potential benefits of integrating information at low levels include leveraging the close temporal correlations and detailed information that may exist. Benefits of integrating information at the conceptual level include identifying ambiguities and hypotheses of meaning that can be used to further refine the understanding of intent.

4.1.1 Synthesizing Semantically Grounded Concepts

The reason for analyzing human hand gestures and facial expressions is to arrive at a set of meaningful concepts that reflect the underlying human intent. At this point in the process, information about human expression is represented as a finite set of primitives (some may be augmented with additional information, such as trajectory of deictic hand gestures). These primitives may not provide value or meaning on their own, but, when aggregated, they can convey specific, meaningful concepts. The primitive integration and semantic grounding module combines sets of plausible primitives over short time frames to identify a small set of reasonable concepts. As illustrated in Figure 4.3, sets of primitives are integrated and semantically grounded using existing expert knowledge. Defining each concept in terms of a few human expression primitives is a task easily achievable directly by hand, while still providing an opportunity for refinement by training using machine learning techniques in the future.

Fuzzy temporal constraints and linguistic rules are used to identify possible concepts given a finite set of previously identified primitives. Each temporal rule defines the relationship between sequential or concurrent primitives from one or more modalities, and linguistic rules define the importance of each primitive to describe a specific concept. An example of a linguistic rule is *If the curling fingers primitive is medium and the cupped finger primitive is high and the palm is facing left and the index finger MCP is facing away from the body, then the glass concept is high*, or *If the mouth is wide open, and the finger grasping primitive is high, and the hand makes a nonvertical movement toward the face, the eat concept is high.* Similar to the fuzzy rule-base used for recognition of static primitives, membership functions can be fine-tuned using labelled training data in a genetic algorithm optimization framework to further improve performance without loosing the benefits of human interpretable knowledge.

It should be noted that the author's approach intentionally places minimal emphasis on formal grammatical structure or the traditional syntax analysis stage. As shown in previous work [85], reasonable results can be obtained solely by matching hand gesture primitives against existing domain knowledge. In this prior research, a manually generated ontology and set of conceptual graphs defined the domain specific knowledge. The accuracy of phrases resulting from this approach was shown to be consistently better than or equivalent to an approach relying solely on low-level recognition [85]. The multimodal approach presented in this thesis is based on a similar basic strategy, where the use of domain knowledge and semantic relationships is leveraged to relax the need for rigorous syntactical analysis. It is an onerous task to define rigid and exhaustive rules of syntax for a single modality, and even more challenging across multiple modalities. Many forms of human expression are not easily described using rigid syntax rules. If meaningful sets of concepts can be successfully identified using loose relationships between primitives and concepts, there is little need to increase the design and computational complexity through rigorous syntactical analysis that is appropriate for only a subset of possible human

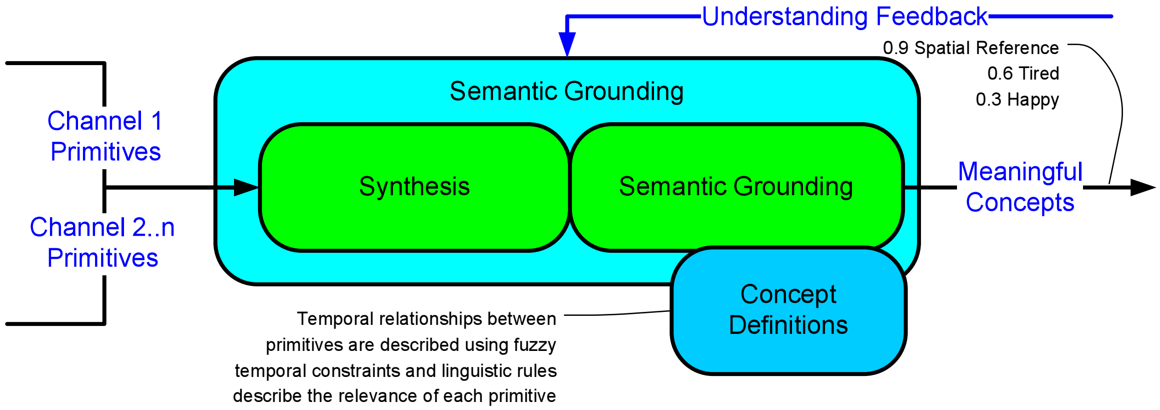


Figure 4.3: Concepts are synthesized from one or more concurrent primitives, semantically grounded using existing concept definitions.

expressions.

Since information about desired human intent can be expressed across multiple modalities in different forms, relationships between primitives and concepts are not specific to any single modality. A finite set of primitives is identified in the recognition module based on a set of observations in a set of channels. In this integration and semantic grounding module, the important information lies not with modalities and channels, but with primitives. The origin of the primitive is no longer relevant. A set of primitives representing a deictic hand gesture may represent a spatial reference concept. The same concept could be described in terms of gaze or specific spoken words. The resulting concept is the same regardless of the originating modality. The level of certainty associated with each primitive, however, may differ from modality to modality.

It is important to note that supplemental parameters provided by primitives can be preserved when forming concepts for future use. One of the more useful supplemental parameters is the trajectory of the finger during a deictic hand gesture. This trajectory information is critical to describe a transparent spatial reference concept, but superfluous in other concepts including the concepts of food, eating, or cooking. Even after integration and semantic grounding, some concepts cannot be fully resolved without incorporating additional information from external sources. For example, spatial references provide little meaning without knowledge of the environment and a means to anchor concepts to physical entities.

4.1.1.1 Synthesis Algorithm

The input to the integration and semantic grounding module is a finite set of recently identified primitives; $\lambda_1, \lambda_2, \lambda_3, \cdots, \lambda_n$, each with associated time intervals: s_i, e_i and a measure representing the quality of fit $Q_1, Q_2, \cdots, Q_n$ between each primitive candidate (static or dynamic) and its associated observations. This set of information varies with time, as new primitives are identified in the observed human expressions, and as additional information becomes available and modifies the quality or duration of previously identified primitives. At each time interval,

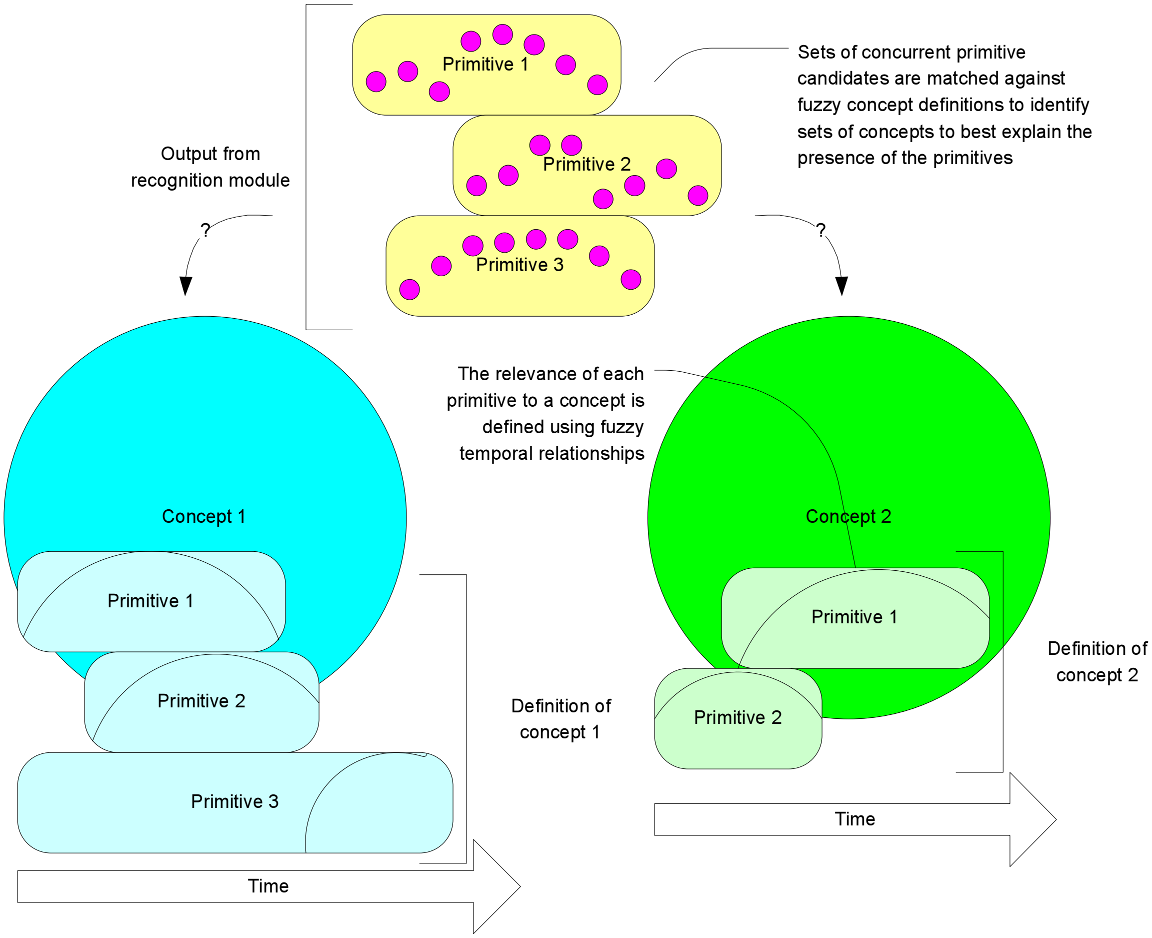

Primitive 1
Primitive 2
Primitive 3
Sets of concurrent primitive candidates are matched against fuzzy concept definitions to identify sets of concepts to best explain the presence of the primitives
Output from recognition module
?
?
The relevance of each primitive to a concept is defined using fuzzy temporal relationships
Concept 1
Primitive 1
Primitive 2
Primitive 3
Definition of concept 1
Time
Concept 2
Primitive 1
Primitive 2
Definition of concept 2
Time

Figure 4.4: Each concept is described by the expression of one or more primitives. The relevance of each primitive may vary across the time interval of a concept, and the relative timing between sequential or concurrent primitives can be important to distinguish one concept from another.

the highest k ranked primitives (at most) are available for integration and semantic grounding. The rank of each primitive is only used inside the recognition module to enforce an upper bound on the number of primitive candidates at the time frame under consideration.

As shown in Algorithm 4.1, concepts are selected to maximize their total relevance over a given duration of time. Since a single concept cannot overlap itself, a dynamic programming approach can be used to select intervals of time to maximize the quality of fit for a given primitive over the entire duration of time. At this point, it is safe to assume the duration of time being considered is a known constant. The approach used to arrive at an appropriate interval of time is discussed along with forming hypotheses of intent in the later portions of this chapter. To identify the most likely time intervals for a single concept, the given approach requires approximately $O(T^3)$ time and $O(T^2)$ space, where T is the number of distinct time intervals under consideration. This process is repeated for all known feasible concepts. Only concepts that include at least one of the provided primitives $\lambda_1, \lambda_2, \cdots, \lambda_n$ are considered. If additional constraints are imposed on

```
Input: Set of primitive candidates with timing information, P
Input: Individual concept C
Input: Duration of maximum time interval, T
Input: Penalty for excessive disjunct time intervals, penalty
GetConceptIntervals(P, C, T, penalty)
// Memoization of quality at different time intervals
for $i$ := $1$ to $T$ do
    for $j = i$ to $T$ do
        // How closely does the concept describe the set of candidate
           primitives observed between time $i$ and time $j$?
        quality(i,j) := Compare( P(i,j), C )
// Calculate optimal value from time 0 through time $i$
for $i$ := $1$ to $T$ do
    $M_C[i] = \max_{j=1 \ldots i}(quality(i, j) + penalty + M_C[i-1])$
// Reconstruct the set of time intervals $I$ for concept $C$ to provide optimal
   quality
j := T
I := {}
while $j > 0$ do
    i := $\operatorname{argmax}_{i=1 \ldots j}(quality(i, j) + penalty + M_C[i-1])$
    I := $I \bigcap \{(i, j, quality(i, j))\}$
    j := i-1
return $I$
```

Algorithm 4.1: Identifying Concept Time Intervals From a Set of Primitives

the quality(...) function, it is possible to reduce the lower bound on time to $O(T^2)$ by using a recursive definition for quality.

The algorithm used to determine appropriate time intervals for each concept relies on a comparison between a subset of available concurrent primitives and the concept under consideration. This comparison function is assumed to normalize the measure of quality using the size of the input to allow the quality of two different time durations to be objectively compared. A penalty is introduced to prevent multiple, unnecessarily small time intervals from being identified. A larger penalty helps to combine multiple small time intervals into a single larger time interval, whereas a small penalty will permit more distinct time intervals.

The problem of comparing a set of candidate primitives against a single concept is similar in nature to a sequence alignment problem. The objective is to find a matching M that assigns each time interval described in each primitive to a corresponding time interval described by a fuzzy set while preserving the relative order of time intervals. No two intervals of time in any primitive can be paired with the same interval of time in the concept. Similarly, each interval of time in the concept definition can only be paired with a single interval of time in a primitive. To enforce relative timing constraints, all concurrent primitives are stretched or compressed across time together.

Concepts are assigned values to indicate the quality or relevance to a given set of candidate primitives. This value is used to rank and determine the most accurate subset of concepts. The accuracy of the match between a concept and a set of given primitives is determined using fuzzy temporal information embedded in each concept. The inference process provides a measure of relevance for each primitive associated with a concept and is used to determine the the concepts that are best described by the given primitives. Primitives not included in the definition of a concept do not impact the quality of that concept, but can still influence its rank with respect to other concepts.

The quality of match given to each concept (c_i) over a given time interval, for a given alignment of the primitives is calculated using a similarity measure between the time profile given by the aligned primitives and the fuzzy temporal constraints given in each concept definition. In cases where observed primitives are repeated and disjunct, the time warping process will still consider the entire time interval. In cases where the concept occurs multiple times, it will be identified when searching for the optimal concept intervals as shown in Algorithm 4.1.

4.1.1.2 Forming a Fuzzy Set Using Primitive Candidate Information

The measure of similarity between aligned primitives and concept information leverages existing fuzzy set theory. Since the information about a primitive is provided as a set of measures of possibilities over potentially overlapping time intervals, this information cannot be considered mutually exclusive, nor can it be considered independent. Several alternatives exist to aggregate information accumulated about primitives over time to maximize the usefulness of the complete set. The alternatives to combine this information follow one of three theories: Bayesian probability theory, the Dempster-Shafer theory of evidence, or the fuzzy set theory. Bayesian approaches require precise information about probabilities, which cannot be guaranteed for the primitives being considered. It is possible to make assumptions about the available information to reduce uncertainty, but these assumptions are difficult to justify, and are only be valid in specific scenarios. A Dempster-Shafer approach handles uncertainty using ranges of probabilities rather than a single probability value. The lower probability bound is given by a measure of belief, and the upper bound by a measure of plausibility. Fuzzy set theory also explicitly represents uncertainty, but with an emphasis on vagueness. Vagueness is one type of uncertainty that defines a level of crispness or precision in the boundaries between sets. An approach based on fuzzy set theory is useful to accommodate the vagueness in human descriptions about concepts while encoding the uncertainty in primitives over time

A fuzzy set is defined on a domain called the universe of discourse, and has values defined by membership functions. The universe of discourse (U) for the set of primitive candidates is a discrete set of times corresponding to the time interval selected in Algorithm 4.1. Membership values define the degree to which each time instant belongs to a specific primitive. Since primitives are not mutually exclusive, a time instant may belong to multiple primitives simultaneously. This concurrency is handled in the fuzzy representation, as the sum of the value of each membership function $\mu_{\lambda_i}(\cdots)$ for all primitives at a given time u_1 is not required to be 1 as it would be in a probabilistic framework. $\mu_{\lambda_1}(u_1) + \mu_{\lambda_1}(u_2) \neq 1$ is as valid as $\mu_{\lambda_1}(u_1) + \mu_{\lambda_1}(u_2) = 1$.

The information provided for multiple primitives is combined using an intersection between the sets, a T-norm. The min T-norm is used for this purpose for two reasons: The calculation is inexpensive, and together with its dual T-conorm ($S(a, b) = max(a, b)$), the relationship $a + b = T(a, b) + S(a, b)$ is met. This relationship is useful when evaluating the similarity between fuzzy sets, as is required when comparing a primitive against the temporal constraints specified in each concept. As shown in Algorithm 4.2, information for a single primitive provides information about its uncertainty for (potentially) overlapping time intervals. This aggregation process uses the pessimistic minimum operator to ensure the certainty of a primitive at any given time is not overestimated.

```
Input: All occurrences $1 \cdots n$ for a given primitive $A$ over the time interval $t_s \cdots t_e$
FormMembershipFunction(A[$1 \cdots n$, $t_s \cdots t_e$], $t_s$, $t_e$)
for $t = t_s$ to $t_e$ do
    $\mu_{\lambda_A}(t) = \min_{1 \leq i \leq n} (A[i, t])$
return $\mu_{\lambda_A}$
```

Algorithm 4.2: Forming A Membership Function for Primitive A over the time interval $t_s \ldots t_e$

4.1.1.3 Measuring Similarity Between a Set of Primitives and a Concept

The algorithm used to compare a set of primitives against a concept uses the similarity between the time profile of each desired primitive and the profiles described by the fuzzy sets included in the concept definition. Each concept is defined in terms of definitions of relative time for each of its primitives, as shown in Figure 4.4. Relative time information between primitives is encoded in the definition of each concept using fuzzy temporal constraints to explicitly capture the vagueness present in primitive timing. The relative timing of primitives is useful to describe a concept such as breakfast in terms of primitives that occur after one another, start at about the same time, end at about the same time, or occur simultaneously. The concept of breakfast can be described as starting with an arcing hand movement toward the face, but during the last part of the arcing movement, the fingers extend and contact the thumb and the mouth opens. The timing between these primitives is important and precise in meaning, but vague in value.

The presence of a concept is determined by the existence and timing of its constituent primitives. A measure of uncertainty about the presence of each concept is similarly determined based on relationships between the concept and its constituent primitives. This measure of uncertainty indirectly compares the similarity of human expressions to a known concept. The similarity of a human expression is based both on the similarity of an observation to dynamic or static primitives, and on the relative timing of these primitives. The similarity between an observation and dynamic or static primitive is calculated in the immediately preceding module, and provided in the form of values between 0 and 1 associated with specific time intervals. As previously alluded, the relative timing of these primitives can be measured using the similarity between fuzzy sets.

The fuzzy sets describing relevant primitives, and the timing information encoded in a concept can be compared in many ways. Obviously, these two fuzzy sets are as similar as possible when they are equivalent, or when

$$\mu_A(t) = \mu_B(t) \forall t \in T, \tag{4.1}$$

where $\mu_A(u)$ is the value of set A at time t, $\mu_B(u)$ is the value of set B at time t, and T is the universe of discourse. A value of 0 is desired when the two fuzzy sets are as dissimilar as possible, and values should represent increasing similarity from 0 to 1. The sup-min compatibility measure is one possibility,

$$\sup_{t \in T} \min(\mu_A(t), \mu_B(t)) = \sup(A \bigcap B). \tag{4.2}$$

Unfortunately, this measure can result in a value indicating the maximum similarity (1) with the existence of only one time value (t) where $\mu_A(t) = \mu_B(t) = 1$. It is important to consider the remaining values across the universe of discourse, even in the case where a perfect match is found for a single value. The maximum similarity value of 1 should be reserved for cases where no closer match is possible at any value of t. The sup-min compatibility measure is considered a partial matching index, not a true measure of similarity between fuzzy sets [106]. A measure of similarity between fuzzy sets should only produce a value of 1 when the two fuzzy sets are identical as shown in Equation 4.1.

Similarity between fuzzy sets can be determined using the symmetric difference between the two fuzzy sets, or by measuring the distance between each fuzzy set when represented as a point in n-dimensional space. A fuzzy set can be transformed to a point using summary information about the set, including its centre of gravity, or centre of area. The symmetric difference is applied in this research to avoid information loss in the transformation of a fuzzy set to a point in n-dimensional space. Comparing the fuzzy set representing a primitive over a specific time interval against a fuzzy set in the concept's definition becomes,

$$\text{similarity}(A, B) = \min\left(\frac{|A \bigcap B|}{|A|}, \frac{|A \bigcap B|}{|B|}\right). \tag{4.3}$$

Now that a value is available to indicate similarity between one alignment of a primitive and its match in a concept, it can be combined with information about the similarity between human expression and the primitive itself to arrive at a measure of certainty in a concept. The certainty of a concept is calculated using a sum of the similarity of its primitives S_c against the relative time information encoded in the concept, weighted by the certainty in each primitive and the area of the corresponding fuzzy set. As shown in Algorithm 4.3, this measure is normalized to account for variances in both the number and relative importance of primitives. $\mu_i(t) = 0$ for values of t where primitive j is not defined in concept C. In cases where the vague temporal constraint information is granular, or defined using discrete time intervals larger than the corresponding information describing primitives ($\mu_j(t)$), the maximum value of the candidate primitive $\mu_j(t)$ over the corresponding discrete time interval is used.

For example, the concept of breakfast can be represented using an upward sweeping motion of the arm toward the face (morning) followed by extended touching digits moving a short distance toward the mouth (eat), while the face transitions between a neutral and shocked expression. The relative timing of these primitives can be significant, and can be represented in the concept definition by including appropriate fuzzy temporal constraints describing a high relevance of the sweeping arm motion primitive early, and a high relevance of the remaining primitives later in the duration of the concept.

```
Input: Set of candidate primitives over specific time interval, P
Input: Concept, C
Compare(P[$t_s \cdots t_e$],C)
$S_c := 0$ // Concept similarity
$N := 0$ // Normalizing factor
// Define universe of discourse for concept definitions as $T = [t_s \cdots t_e]$
for each primitive instance i in C do
    for each candidate primitive instance j of the same type as i do
        // Calculate similarity
        intersect := $\sum_{t \in T} \mu_i(t) \wedge \mu_j(t)$
        $S_p$ := intersect / max($\sum_{t \in T} \mu_i(t), \sum_{t \in T} \mu_j(t)$)
        $S_c := S_c + S_p \times \sum_{t \in T} \mu_i(t)$
        $N := N + \times \sum_{t \in T} \mu_i(t)$
$S_c := S_c / N$
return $S_c$
```

Algorithm 4.3: *Comparing A Concept Against Available Primitives*

A useful side-effect of increasing the granularity from a large (but bounded) set of primitives to a few concepts is the relatively straightforward extraction of cues for plausible knowledge-based temporal segmentation points. The duration between adjacent concepts can be used to measure a *timeout* period that may be used in future research to separate two different phrases or intents as determined in the subsequent understanding module. It is important to note that these segmentation points will vary depending on the set of concepts selected. The duration between two adjacent concepts may not be adequately described by a third concept, but often still contains recognized primitives that are still not associated with a selected concept. The information about the alignment of concepts with specific time intervals is valuable to indicate a lack of meaningful human expression or simply unexplainable observations.

4.2 Forming Hypotheses Of Intent From Concepts

> **understanding,** *n.*
> **1.** a mental grasp
> **2. a:** the power of comprehending; *especially*: the capacity to apprehend general relations of particulars **b:** the power to make experience intelligible by applying concepts and categories
>
> — Merriam-Webster Online Dictionary

It is important to explain here why the distinction between the term *understanding* and recognition is important. Recognizing an observation provides an explicit label or identifier for future use. Understanding, on the other hand, requires observed information to be associated with *a priori* knowledge, not necessarily by providing an explicit label, but with relationships to relevant knowledge. It is these relationships that provide meaning to an understood concept, unlike an explicit label that does not require a meaning. Understanding allows relationships and relevant concepts to be inferred about the observed human expressions. In this research, the process of understanding deals exclusively with sets of concepts produced from the integration module. This process is at a sufficiently high level to be explicitly decoupled from any one specific modality or combination of modalities.

Meaningful sets of concept candidates produced during integration are brought together to form hypotheses of intent in the understanding module. A priori knowledge is used to guide this process, including both predefined domain specific knowledge, and context or knowledge about recent human communication and robot state or current action. Up to this point in the process, only information about individual concepts and primitives, or the building blocks of human expression have been used. Primitives and concepts are generally domain independent, and applicable in any context or interaction scenario. If specific, relevant interaction scenarios are known, this knowledge can be valuable to extract meaning or understand the underlying purpose of human expressions.

In human-robot interaction, the robot, as a target of human expression, is limited to carrying out a finite set of high-level actions. Numerous variants of each type of action may exist, but these can be grouped together and considered the same high-level action. Multiple alternate paths to follow or different grasping techniques to apply when retrieving an object are results of internal robot decisions, and simply alternate approaches to carry out the same action as desired by the human expression. Each high-level action or interaction scenario must be represented so it can be used to associate meaning with sets of previously identified candidate concepts. Knowledge representation is a large area of research on its own, with many alternate forms. In this research, a form of conceptual graph is used to represent knowledge about scenarios specific to the interaction domain and robot capabilities.

Traditional conceptual graphs provide a useful human interpretable framework to represent and reason about concepts and relationships between concepts. These graphs include sufficient information to represent any first order logic expression. Since a relatively straight-forward transformation exists from first order logic to conceptual graphs, one might wonder why first order logic is not directly used. Unfortunately, formal logic is rigid and precise. It is difficult to generate accurate logic about the robot's environment or human-robot interaction scenarios and also take advantage of information about uncertainty when using first order logic. A combination of fuzzy set theory with conceptual graphs, or fuzzy conceptual graphs allows this uncertainty to be included directly in the representation of domain specific knowledge [107, 108]. The remainder of this section discusses how domain specific knowledge is represented in the form of fuzzy conceptual graphs, and how approximate reasoning is performed over these graphs to form hypotheses of intent from observed human expressions.

4.2.1 Representing Knowledge For Human-Robot Interaction

Effective communication relies on a shared, common base of knowledge. This fundamental knowledge includes what many people consider common-sense or trivial facts. Nevertheless, these common-sense or trivial facts are essential for any participant to provide the context or background required to understand complex human expressions during interaction. Language itself is a form of shared knowledge. When two participants attempt to communicate without a common language, negligible information will be exchanged until both participants invest the substantial effort and time required to first exchange the necessary language information. Similarly, two participants from different cultures may interpret body language differently. Without a common base of knowledge, participants will either fail to understand one another, or misinterpret the underlying intent.

A shared base of knowledge is especially important when one participant is a robot or automated device. The base of knowledge must be explicit and as comprehensive as possible over the domain of interaction, as devices are currently incapable of extrapolating or inferring meaning as efficiently as humans. Ambitious long-term projects exist to help tackle the elusive natural machine understanding problem by encoding human common-sense knowledge for use in devices [109]. This effort to encode human common-sense knowledge is valuable, but determining when the set of common-sense knowledge is complete can be considered as intractable as the problem of natural machine understanding itself.

Methods to encode knowledge have been developed and analyzed since the beginning of philosophy. An ontology is an attempt to define the essential components of concepts. Defining concepts, or placing these concepts into categories, is an approach to unambiguously describe the essence of each concept. Unfortunately, describing what an item is, or which parts of a concept remain the same after a change, is an ongoing philosophical challenge. Limiting the scope or domain of knowledge simplifies the process of developing a useful ontology that can be re-used while being aware of the domain specific restrictions.

When focusing on human-robot interaction, the vast scope of common-sense knowledge can be reduced to a minuscule subset containing only the knowledge relevant to the interaction domain of the robot. Rather than tackling the vast domain-independent knowledge representation problem, the base of knowledge shared between the human and robot participants in human-robot interaction can be safely specialized to the relevant domain. Robots, and devices in general are designed with a purpose. These devices are designed to carry out a specific set of tasks, which means the basic knowledge about these tasks or the capabilities must be known during the design process, even if they are not explicitly documented. These tasks or capabilities can be used to constrain the scope of human domain knowledge that must be imparted on a robot for efficient human-robot interaction.

Knowing the scope of knowledge that must be shared only places bounds on the size of the representation, and does not help specify the form in which this knowledge is represented. Knowledge has been represented for several decades using frames [73]. A frame represents a concept type, and defines a set of slots that help to narrow the concept type to a specific instance of a concept. Labelled graphs are also a popular tool to represent knowledge. Labelled graphs, unlike frames and slots, provide flexibility and can leverage existing graph theoretic techniques for knowledge manipulation. Non-trivial relationships between concepts can be represented easier in graph form than in frame form. The phrase *a robot is between the door and the table* can be easily represented using a graph with four vertexes and three edges, but is difficult to represent in frame form since the relationship *between* is associated with three different concepts. In the artificial intelligence community, labelled graphs for knowledge representation are commonly called semantic networks.

The knowledge representation selected for use in this thesis is a based on a labelled graph based formalization, the conceptual graph [110]. This form of knowledge representation is used to encode domain specific knowledge, both for its ease of visualization and maintenance, and for the simplicity in extending the formulation to relax traditional rigid logical constraints. Conceptual graphs were defined by Sowa, based on Charles Sanders Peirce's work and a reformulation of semantic nets. Reasoning with conceptual graphs can be performed using standard graph operations, eliminating the need to handle specific design and implementation considerations for an alternate representation such as a frame based approach.

A conceptual graph is a bipartite directed graph used for knowledge representation [110]. One set of nodes represent concepts, and the other set relations, often between concepts. Each concept has attributes describing its type and a referent or instance. A concept representing the mobile robot named mag2 may have an attribute describing the type *mobile robot* and a referent for the specific instance *mag2*. Each relation has attributes to describe its type and the number of concepts it must be connected to with arcs. Each arc belongs to a relation node and is numbered based on the relation node type. A binary relation such as *agent* requires arcs two two concepts, from a concept to its agent. A simple conceptual graph illustrating relation nodes, concept nodes, arcs and node labels is illustrated in Figure 4.5.

Using just the structure of a simple conceptual graph without supporting knowledge is not very helpful on its own. Each concept and relation type belongs to a lattice structure of related concepts or relations. This structure classifies concepts between *Entity*, the universal type under which everything is a subtype, and *Absurdity*, the absurd type to which everything is a supertype. Two concepts may be related to one another using this ontology of known concepts to identify subsumption relations for generalization or specialization of a specific conceptual graph. Similarly,

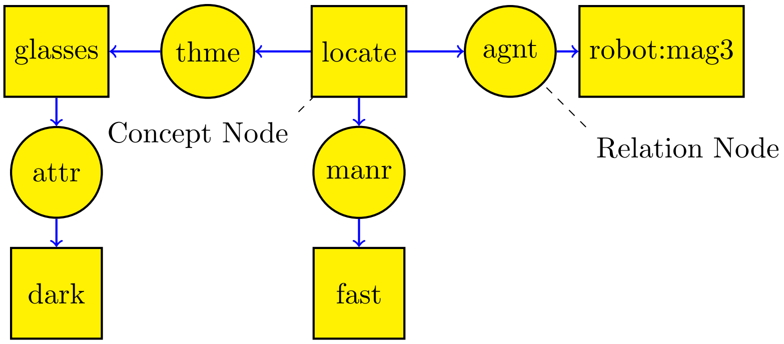


Figure 4.5: Simple conceptual graphs are bipartite labelled graphs with concepts and relations. Concepts are described with a type and an optional referent to indicate a specific instance.

each relation type belongs to an ontology to describe relationships between any two relation types.

4.2.2 Reasoning With Knowledge For Human-Robot Interaction

Formal reasoning can be a computationally intensive task, requiring accurate and rigorous logical rules to produce reasonable results. To take advantage of the developed fuzzy knowledge base with less rigorous but practical rules, the author uses a low-complexity and intuitive conceptual graph based analogical reasoning approach [111].

One of the strengths of a graph based representation for knowledge is the simplicity of reasoning. Reasoning on conceptual graphs is performed by applying graph operations. Six fundamental operations exist for reasoning with conceptual graphs; specialization (generalization) , copy (simplify), an restrict (unrestrict). Specialization produces a graph that is logically more specific or implied from the original graph, and generalization is the inverse. Similarly, the copy operation results in a duplicate conceptual graph without changing its meaning and simplify is the inverse. Restrict changes the meaning of a conceptual graph by modification of referent or type, and unrestrict is the inverse operation. Examples of these six fundamental operations are illustrated in Figure 4.6, Figure 4.7, and Figure 4.8.

At this point in the process, a set of ranked concept candidates over specific time intervals is available. These concept possibilities are based on information provided from each modality specific processing module. A set of known scenarios is also available in the form of domain-specific conceptual graphs. These known scenarios are used to extract meaning from the available sets of concepts and form reasonable hypotheses of intent. The standard conceptual graph operations are used to manipulate domain specific knowledge and find analogies that can best account for the available set of concept candidates.

Advantages of this approach to both integrate multiple concepts and understand by finding analogies rather than direct application of formal logic include simple initial setup and scenario modification without need for precise and consistent formal logic. One of the major issues with any domain specific knowledge base is the ability of a human to understand and maintain the knowledge itself. The more difficult the representation and reasoning scheme is, the less likely it

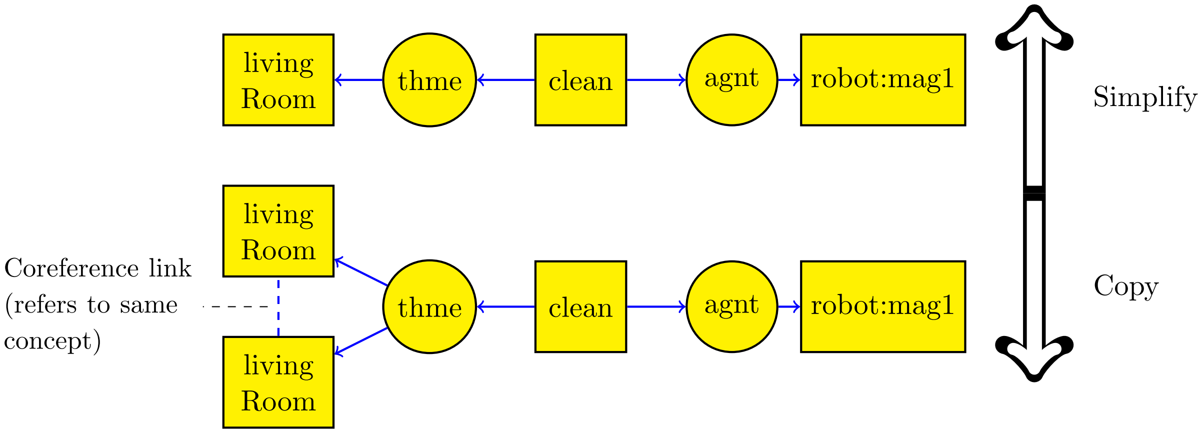


Figure 4.6: Simplification is used to reduce redundancy in a conceptual graph. The opposite operation, copy, can be used to increase redundancy. Duplicate concepts are indicated in the conceptual graph and associated with one another using a coreference link.

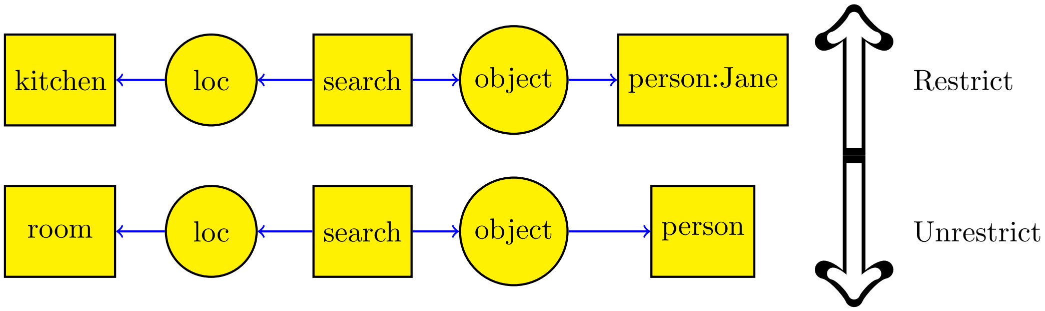


Figure 4.7: Restriction or specialization uses a general statement in the form of a conceptual graph to make an implication about a more specific statement. The opposite operation to make a general statement when given a specific instance is unrestriction or generalization.

will be accurately maintained. Other advantages include the low average computational complexity of reasoning and ability to operate even with inconsistent or incomplete information. General rules and past examples can be automatically extended and applied to new scenarios as long as reasonable analogies exist. Obviously, without the rigorous and accurate logical rules of first order logic, the hypotheses generated by this approach will still contain some level of uncertainty and are not guaranteed to be consistent with all information extracted from observed human expressions.

Alternative hypotheses of intent are evaluated by both their consistency against existing knowledge and the level of confidence in the concepts derived from human expression. A hypothesis of intent consistent with the current state and capabilities of the robot has a greater possibility of being the desired human intent than a hypothesis that is in conflict with known information. Similarly, a hypothesis of intent based on concepts with high measures of confidence is considered a more plausible interpretation of intent than a hypothesis based on identified concepts with low

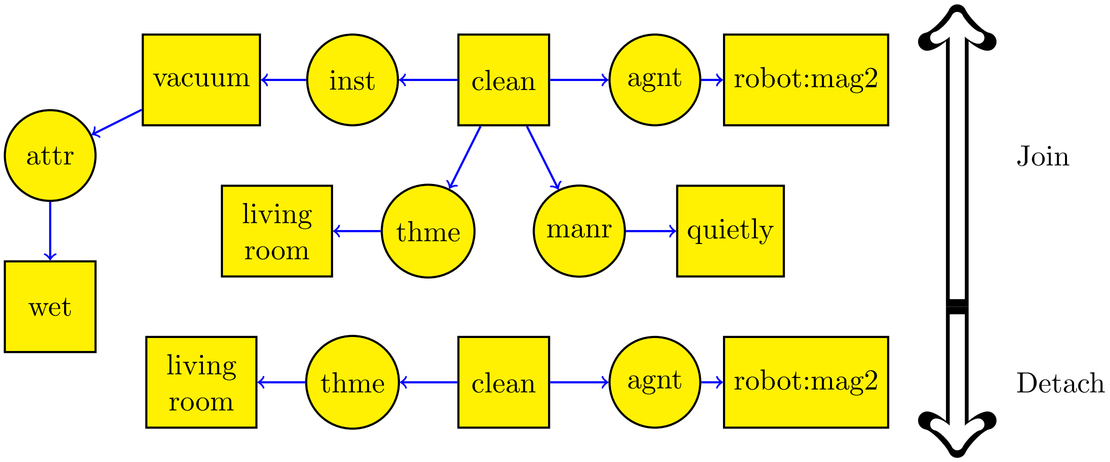


Figure 4.8: Relevant conceptual graphs can be joined to combine logical statements with common elements. Similarly, a large conceptual graph can be divided into multiple disjunct graphs using the detach operation.

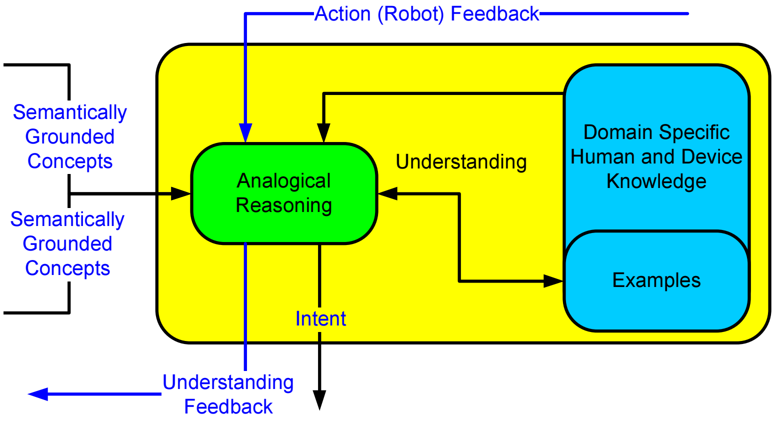

Action (Robot) Feedback
Semantically Grounded Concepts
Semantically Grounded Concepts
Analogical Reasoning
Understanding
Domain Specific Human and Device Knowledge
Examples
Intent
Understanding Feedback

Figure 4.9: An analogical reasoning approach to understanding is applied to combine sets of concepts provided from each modality specific module into a reasonable interpretation of human expression. Domain specific knowledge is represented and manipulated using conceptual graphs.

measures of confidence.

4.2.3 Finding Analogies

The existing knowledge base consists of multiple independent fuzzy conceptual graphs, each representing a known scenario or prior example of communicated intent. The set of concept candidates is arranged to minimize the semantic distance between the complete set of concept candidates and each existing case or prior example in the knowledge base. The most similar conceptual graph is then selected as the basis from which overall intent is inferred.

The semantic distance between concepts is calculated using a measure of similarity between concept attributes, including their types and referents. The similarity between two concept types is based on both the distance to a common general concept, and on the specificity of each concept. The distance to a common general concept provides information that can be used to show that the moon and sun are more similar than a pencil and smoke. Specificity is important to include in this measure of similarity to prevent very specific concepts from being penalized by their depth in the concept ontology and their proximity to the absurdity concept ($\perp$). A HB pencil and a ball-point pen are no less similar than a pencil and a pen, even though the distances to the closest common general concept for the HB pencil and ball-point pen may be longer.

As shown in Equation 4.4, the measure of similarity $S(a, b)$ between two concept types a and b is a value between 0 and 1, where 1 is the most similar, and 0 the least similar. c is the closest common more general concept to a and b, and the distance between two concepts is given by the function $d(x, y)$ ($d(x, x) = 0$, $d(\mathit{entity}, \mathit{absurdity}) = m$). $2m$ is a normalization factor based on the maximum depth of the concept lattice, m.

$$S(a,b) = 1 - \frac{(d(a,c) + d(b,c))(d(a,\perp) + d(b,\perp))}{2m(d(a,\mathit{entity}) + d(b,\mathit{entity}))} \tag{4.4}$$

The overall similarity between a set of concept candidates and an existing case is calculated using the sum of individual concept similarities, weighted by their importance to the specific case. This measure of overall similarity is reduced based on the remaining set of concept candidates that could not be associated with concepts in the case under consideration. A penalty is also applied based on the importance of remaining unmatched concepts and incomplete relations in the existing conceptual graph.

The measure of overall similarity given in Equation 4.5 defines an objective function for this combinatorial optimization problem,

$$f(C, G, \pi) = \frac{\sum_{(c,a)\in\pi} \min(S(c,a), I(G,a))}{\sum_{c\in G} I(G,c)} - \sum_{b|(b,i)\notin\pi\forall i} c_b. \tag{4.5}$$

The first component of this function includes the similarity between matched concepts the the importance of these matched concepts to the given case. This information is normalized to compensate for differences in sizes and measures of importance in alternate conceptual graphs. Conceptual graphs that contain unmatched but important concepts will be penalized more heavily by the large denominator than graphs with matched or less important unmatched concepts. Concepts that are essential to describing the given scenario have an importance of 1. Concepts that are optional, but add some value to the scenario have low importance. An importance of 0 can be used for a concept that neither adds nor detracts from the given domain-specific scenario.

The second component of this objective function captures the remaining unmatched concept candidates. Concept candidates that are not mapped to concepts in the conceptual graph penalize the objective function proportional to the provided measure of quality from the integration module. A high quality concept candidate penalizes the objective function more than a low ranked, borderline concept candidate.

The set of concept candidates is given by C. G is the set of concepts in the conceptual graph. The mapping between concept candidates and concepts in the conceptual graph is defined by π,

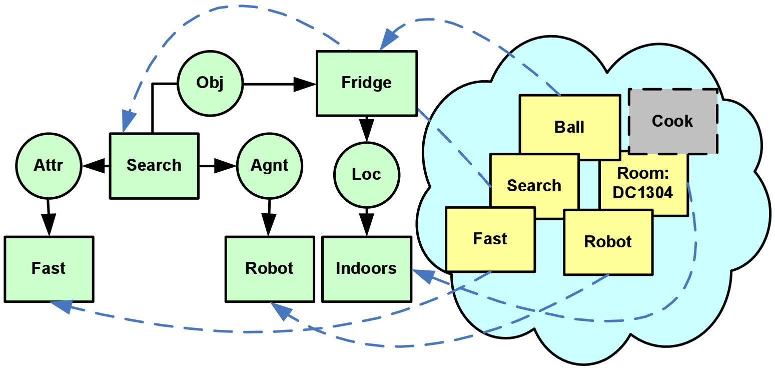


Figure 4.10: A set of concept candidates, shown on the right, is matched against prior scenarios to find a reasonable analogy to explain the observed human communication. Each concept on the left contains information about its relevance to the scenario being described.

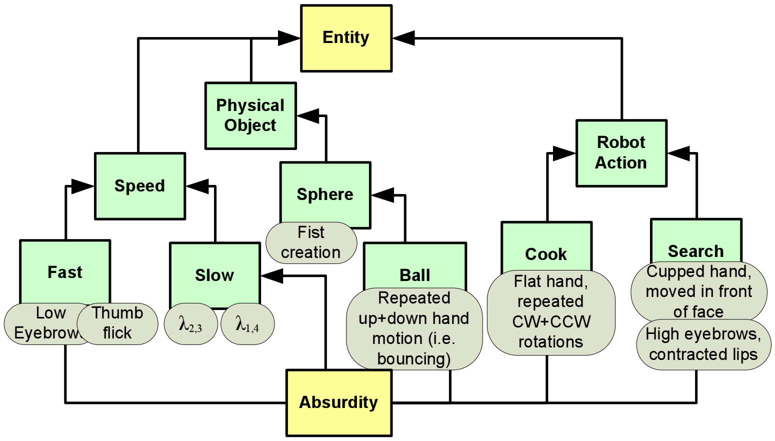

Entity
Physical Object
Speed
Sphere
Robot Action
Fist creation
Fast
Slow
Ball
Cook
Search
Low Eyebrow
Thumb flick
$\lambda_{2,3}$
$\lambda_{1,4}$
Repeated up+down hand motion (i.e. bouncing)
Flat hand, repeated CW+CCW rotations
Cupped hand, moved in front of face
High eyebrows, contracted lips
Absurdity

Figure 4.11: The domain specific ontology defines relationships between concepts. Distances between concepts in this ontology are used to define a measure of similarity between concepts. Each concept is defined by a set of concurrent primitives associated using temporal constraints and fuzzy rules.

and the measure of importance of a concept a to the scenario described in the conceptual graph G is given by the function $I(G, a)$, where $0 \leq I(G, a) \leq 1$. c_b is a measure of the quality of match for concept b as provided from the integration module.

4.2.3.1 Searching For The Best Analogy

The problem of finding the best analogy can be modelled as a combinatorial optimization problem. A combinatorial optimization problem consists of multiple solutions with associated values defined with an objective function. The goal is to find a solution that minimizes (or maximizes) the objective function. A naïve approach may evaluate the objective function on all solutions and simply select the maximum, or arrange solutions in monotonically nondecreasing order by their objective function values. The naïve or brute-force approach may work on simple or small problem, but, unfortunately, many real combinatorial problems have a nontrivial number of solutions. The vast number of solutions in problems with nontrivial solution landscapes prohibits an exhaustive search from finding a global optimum in reasonable time.

When given a set of candidate concepts, C, and a single previously defined scenario, G, a solution consists of a mapping between each candidate concept and a concept in the existing scenario. Considering only the n_c candidate concepts and n_g concepts in the existing scenario, there are

$$\frac{n_c!}{(n_c - n_g)!} = n_g! \binom{n_c}{n_g} \tag{4.6}$$

possible mappings (assuming $n_c \geq n_g$ without loss of generality, and duplicate concepts do not exist in either set). Since it is also possible for a concept to be deleted, subsets must also be considered. There are no more than

$$\sum_{i=1}^{n_g} \binom{n_c}{i} = \sum_{i=1}^{n_c} \binom{n_c}{i} - \sum_{i=n_g+1}^{n_c} \binom{n_c}{i} \tag{4.7}$$

$$= 2^{n_c} - 1 - \sum_{i=0}^{n_c - n_g - 1} \binom{n_c}{i} \tag{4.8}$$

nonempty subsets of G to consider. If each concept can also be altered to become any of the $n_o - 1$ other known concepts in the domain specific ontology in addition to simply being deleted, the maximum number of nonempty mappings becomes

$$\sum_{i=1}^{n_g} \binom{n_c}{i} (n_o + 1)^i \in O(n_c! n_o^{n_g}). \tag{4.9}$$

Since not one, but a set of n_s scenarios must be evaluated to find all solutions, the number of solutions is bounded by $O(n_s n_o^{n_g} n_c!)$. The upper bound on the number of solutions to this combinatorial problem grows faster than a polynomial function of either n_c or n_g, therefore, evaluating every solution is impractical for any nontrivial set of concepts.

This problem can, in fact, be rephrased as a type of subgraph isomorphism problem. If each subset of n_g concepts in the set of n_c candidate concepts are considered fully connected graphs, the objective becomes a minimization of the cost of modifications required to the existing knowledge in G to find a subgraph isomorphism of the graph formed by the set of candidate concepts. The subgraph isomorphism problem is a known NP-complete problem [112]. Finding approximate matches, or correcting errors in one graph to match the other only adds to the complexity of this problem.

Due to the large solution space, and complexity of finding the optimal fit between concept candidates and existing domain knowledge in the form of fuzzy conceptual graphs, an approximation algorithm is necessary to obtain reasonable solutions in reasonable time. Using an approximation algorithm cannot guarantee a global optimum in reasonable time or space, but can be useful to provide a suboptimal solution. Timing is a critical factor in any form of human interaction, including everything from telephone conversations to the focus of this thesis on human-robot interaction. The maximum tolerable latency during human interaction varies widely between applications and studies, including some recommending maximum latencies of 50ms [113], and others indicating variations are more important, recommending maximum standard deviations of no more than 82ms [114]. Regardless of the actual limit on reasonable latency, it is reasonable to state that excessive latency hinders the effectiveness of human interaction. If a robot spends an excessive length of time searching for a global optimum without some form of feedback, the human participant may leave and may even consider the robot rude for not adhering to socially acceptable interaction protocols.

Several relevant optimization algorithms can be applied to solve this optimization problem, including simulated annealing, evolutionary, ant colony, and Tabu search algorithms. All of these algorithms attempt to balance the exploration of new areas of the solution space against exploitation of previously found solutions.

Simulated annealing uses a temperature parameter similar to real-world annealing. An initially high temperature emphasizes diversity and exploration using a probabilistic scheme. As the temperature is reduced, the emphasis shifts from exploration to exploitation of existing solutions in an attempt to converge on a global optima. Evolutionary approaches also explore the solution space probabilistically using large populations of individuals that may represent multiple solutions. Existing solutions are leveraged by combining fit individuals or good quality solutions, and removing less fit individuals. Mutations help to prevent premature convergence. Ant colony algorithms are similar, in that they involve multiple individuals in the search through the solution space. Individuals, or ants in this case, explore the solution space randomly and leave pheromone trails on their return trips if a good solution was encountered. Subsequent ants have a higher probability of following an existing pheromone trail than a new, unexplored path. This form of positive reinforcement helps to exploit and improve on previously found solutions. Pheromones decay over time to emphasize good solutions, or regularly travelled paths. The strength of an ant colony approach is in its ability to dynamically adapt to changes in the solution space without restarting the algorithm. The Tabu search differs from the three previously described algorithms, in that it explicitly leverages both a short and long term memory of visited solutions [115]. Rather than relying heavily on probabilistic movements throughout the solution space, the Tabu search algorithm can be considered an augmented, deterministic local search.

Although any of these approximation algorithms can be applied to the problem of finding a good analogy, the Tabu search is most suitable, as there is a good level of structure in the domain specific knowledge base that can be easily leveraged in a Tabu search. Evolutionary approaches rely on crossovers between individuals to exploit previous solutions, but it is difficult to take advantage of the existing ontology of concepts during this crossover process. Simulated annealing, on the other hand, relies on probabilistic movements and cannot easily leverage information about similar solutions based on the structure of the solution space. Use of a Tabu search does not exclude these other, more probabilistic algorithms. Evolutionary, or simulated annealing approaches can be used to augment a Tabu search by providing initial solutions, just as a Mimetic algorithm combines a genetic algorithm with a local search.

The Tabu search technique [115] is used to find solutions since this deterministic algorithm can take advantage of the structure of conceptual graphs during the search. Other benefits of the Tabu search include its ability to adapt to accommodate the solution landscape of the problem. Adaptation to the solution landscape is influenced by several parameters including the Tabu list length, type, and aspiration criteria. As shown in Algorithm 4.4,

```
Input: Neighbourhood function N(...)
Input: Aspiration criteria A(...)
Input: Objective function f(...)
TabuSearch( N(), A(), f() )
S(0) = Initial solution
i = 0
T = {} // Start with an empty tabu list
while $i \leq$ max iterations and not stalled do
    i = i + 1
    S(i) = $-\infty$
    for each neighbour $j \in N(S(i-1))$ do
        // If solution is already in Tabu list,
        if $f(j) \in T$ then
            // If aspiration criteria met, override Tabu list
            if A(j) then
                S(i) = j
        else
            // Find the best neighbouring solution
            if $f(j) \geq S(i)$ then
                S(i) = j
    // Update the Tabu list to keep track of visited solutions
    T = $T \bigcup \{S(i)\}$
    // Prune tabu list - remove expired / old solutions
return S(i)
```

Algorithm 4.4: The Tabu search is used to search for good analogies by maximizing the quality of a mapping between a set of candidate concepts and existing domain specific knowledge in the form of conceptual graphs.

Since the Tabu search is a deterministic optimization algorithm, the initial solution has a significant impact on its performance. A poorly selected initial solution may prevent the search from finding the global optimum. In this research, an initial solution is selected by first ordering the concepts from the conceptual graph by decreasing importance and specificity. The closest concept candidate is then found for each concept in this list, using each concept candidate only once, and disregarding graph connectivity. Alternate initial solutions include random assignment, or use of a stochastic approximation method to generate initial solutions.

Other parameters required to find a solution using the Tabu search include definition of a local neighbourhood, and management of a Tabu list. In this human expression understanding application, neighbourhoods are defined based on changing the mapping between concept candidates and existing concepts in the conceptual graph. The neighbourhood for each concept candidate includes any concept node connected by a single relation in the conceptual graph. This constraint to map only concepts to concepts ensures the resulting graph is a valid bipartite conceptual graph. The neighbourhood also includes mapping a concept candidate to *empty*, resulting in an unmatched concept candidate. The change in cost to each neighbouring solution is determined by subtracting the contribution of the new mapping from the contribution of the current mapping using the previously defined objective function.

A traditional recency based first-in-first-out Tabu list management method is used during the search. This approach ensures recent solutions are not repeated, helping to balance exploration of the solution space against exploitation of the local solution landscape. Separate lists are maintained for concepts and relations since these two node types have very different characteristics. In general, fewer relation types exist, but with higher frequency than the typically more diverse concept types in a given conceptual graph. A recently used concept candidate is not used in another move until it is no longer in the list or the aspiration criteria is met. Both of these Tabu lists can be overridden if a neighbouring solution is better than the best solution found so far.

4.2.4 User Disagreement

Forming an accurate hypothesis of human intent is difficult even between two people with decades of communication and interaction experience. In cases where the target device arrives at an incorrect hypothesis, it is important to provide a mechanism for a human participant to correct the problem. Human feedback is used to refine the understanding process and compensate for incorrectly identified hypothesis of intent. Unfortunately, up to this point in the process, human expressions are only being perceived and processed to find good analogies, but the purpose of the expressions is not explicitly identified.

Even after finding a reasonable analogy that provides a good explanation for perceived human expressions, some ambiguity may remain about the basic purpose of the human expression. A decision must be made to determine whether the intent is to provide an informative statement, ask a question, or issue a command. In traditional written or spoken English, an informative statement is normally provided as a declarative sentence. Identifying informative statements can be useful in future work to dynamically update domain knowledge from observed human expressions. Simple questions in written English are provided as direct interrogative sentences, and are clearly identified with a question mark. Distinguishing enquiries from informative statements and commands is important to prevent inadvertently carrying out an action rather than answering a question or accepting new high-level information.

This research currently provides limited support to distinguish between direct negative commands or negative informative statements and asserting positive commands. Human feedback can be provided when the user does not agree with the interpretation of intent, and can arrive in the form of a correct and possibly alternative expression of the same intent, or in the form of a negative expression. Since a positive user correction is simply a new user command, there is no need to take a special course action when receiving a positive command correction. Unlike positive commands, direct negative statements provide the device with information about the correctness of its selected course of action. A direct negative statement is used to refine prior interpretations of intent by discounting the most recently selected hypothesis of intent. Negative user feedback is detected when a specific case in the knowledge base is found in the selected hypothesis of intent (stop). The rank of the most recent analogy is adjusted to expose alternate previously discarded analogies.

Chapter Five

Leveraging Conflict

> Difficulties are meant to rouse, not discourage.
> The human spirit is to grow strong by conflict.
>
> — William Ellery Channing

Conflicts arise from disagreements between multiple sources, including everything from face-to-face disagreements to inconsistent information between components of multimodal expression. These disagreements can form from a range of origins, including incompatible information, opposing goals, and flawed models. Although conflict is often associated with groups of people sharing opposing viewpoints, parallels can be drawn between between human conflict and internal device conflict. Conflicts can arise from any information source or decision making process, regardless of whether a person or a robot is involved. In any real-world natural human communication scenario, conflict is unavoidable, and can in fact be an important aspect of communication itself. This chapter focuses on conflict in multimodal human-machine interaction, discussing how conflict can be detected and leveraged throughout the presented multimodal architecture to help refine hypotheses of desired human intent.

Conflict is inevitable in natural language and human expression in general. Unfortunately, the term *conflict* is often used with a negative connotation. Without conflict, there would be no challenge, no confrontation, and few challenges to successfully resolve or overcome. Rather than avoiding, ignoring or trying to reformulate conflict, it should be embraced as is; an integral and valuable component of interaction. The importance of not ignoring conflicts can be illustrated in an aviation scenario, where the pilot must actively search for conflicts between multiple sources of information. If the pilot relies on a single indicator such as an altimeter, and ignores or does not look for conflicting information from other sources (out the window), the pilot may not know when the altimeter is faulty. Although the outcome of a pilot looking solely at a faulty altimeter may be more catastrophic than the outcome of ignoring conflict when understanding human expression, it is beneficial to search and take advantage of conflict.

Since conflict is unavoidable during natural human-robot interaction, the author's approach explicitly identifies and embraces conflict to further improve the machine understanding of human intent. This approach leverages complementary conflict analysis and resolution techniques to bring human-robot interaction one step closer to transparency, a step toward achieving the simplicity that currently exists in human face-to-face interaction.

Conflict arises in almost every field of study, including philosophy, engineering, sociology, economics, linguistics, law, and politics. The multidisciplinary nature of this topic leads to many different definitions and perspectives on conflict itself. In human-robot interaction, conflict can be defined as incompatibilities or inconsistencies between two or more sources of information. These sources of information include internally derived goals and hypotheses, and external input from both the direct participant in the interaction process and perception of the remainder of the robot's environment. For the purposes of this research, it is assumed that the external environment and domain specific knowledge are consistent. This is a reasonable assumption as long as the knowledge base is constrained and the operating environment is in the real world, not a simulated or theoretical environment.

The challenges that must be addressed to successfully embrace conflict in human-robot interaction are discussed in the remainder of this chapter, including an overview of conflict sources, detection, analysis, and resolution. Challenges are presented with a focus on a natural multimodal human-robot interaction framework, however, the same challenges exist in almost any field of human-machine interaction. Recognizing the importance of conflict in modern human-machine interfaces and explicitly embracing conflict is essential before a truly intuitive and natural human-machine interface can be realized.

5.1 Sources of Conflict in Human-Robot Interaction

Identifying the origins of a conflict, or classifying conflict in terms of its source can provide valuable insight to help address the conflict. Since different forms of conflict can exist across different sources of information in human robot interaction, conflicts can be classified by their source. Available sources can be described in terms of three general scenarios, as shown in Figure 5.1. The first scenario, *mode conflict* is the conflict between two or modes used for human expression. The second scenario, *temporal conflict* occurs between concepts expressed over time or between the longer term context and one or more shorter term concepts, and *action conflict* occurs between the action the robot carries out and the intent of the human participant. An example of mode conflict is when a person nods his or her head in agreement while uttering a negative response, or when hands provide a deictic gesture indicating one location while gaze or verbal sources indicate a different location. Action conflict may occur when the robot is currently moving toward a closed door, and a command is given to close the door. Similarly, temporal conflict may occur when a command is first given to travel toward the north wall, followed by a command to examine the chair directly to the east. Each of these sources of conflict present unique challenges that must be considered to successfully resolve inconsistencies.

Alternate methods to classify conflicts include classification based on the technique used for negotiation and resolution rather than the origin of the conflict. A classification approach based on negotiation and resolution is more applicable to human conflicts than human-robot conflicts, as it cannot be determined before progressing beyond detection of a conflict. Before discussing conflict resolution approaches, it is important to explore the sources and types of conflict that may arise in human-robot interaction in detail. Examining the different types of conflict helps identify the relevant components in the presented architecture that can be used for subsequent detection and resolution. A symbiotic relationship exists between conflict and understanding, as some types of conflict may not be detected before a human expression is understood. Similarly,

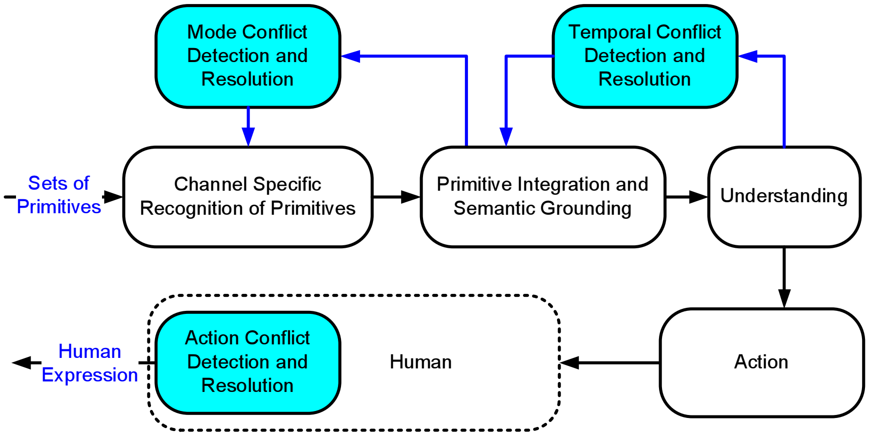

Mode Conflict Detection and Resolution
Temporal Conflict Detection and Resolution
Sets of Primitives
Channel Specific Recognition of Primitives
Primitive Integration and Semantic Grounding
Understanding
Action
Human Expression
Action Conflict Detection and Resolution
Human

Figure 5.1: Conflicts are monitored between primitives, between concepts, and between the action performed and the originating human intent.

the information available from a conflict can be valuable to refine a previous understanding. Unfortunately, the dynamic nature of human expression limits the options available to acquire redundant measurements of the same event. It is extremely difficult and unlikely for a person to express the same event in exactly the same manner twice. Redundant information often exists at a higher level, including after recognition or understanding. The following sections discuss the major sources of conflict in human-robot interaction to illustrate the importance and integral relationship between conflicts and understanding in the closed-loop approach of the presented architecture.

5.1.1 Mode Conflict

Natural human expressions are conveyed across multiple modes and modalities. Unfortunately, what is expressed in one mode may not agree with the information expressed in another. It has been shown that people automatically attempt to compensate for conflicts between some modes based on prior knowledge. Perhaps the most obvious example of this was demonstrated in the McGurk phenomenon. This phenomenon occurs when conflicting sound and vision cues are present [116]. A form of mode conflict is present in the McGurk phenomenon, where a single audio syllable is synchronized with multiple visual movements of the mouth that do not necessarily correspond with the synchronized audio syllable. Even though the same audio stimulus is provided in each combination, people from a wide range of ages and language backgrounds will subconsciously detect and resolve this form of mode conflict. The same audio syllable *ba* may be perceived as *da*, *va*, or *the* when combined with conflicting visual information. It is interesting to note that the human mind can transparently select one of the modalities or combine information from both sound and vision to produce a new syllable as a compromise between the conflicting sources.

At a higher level, a complete verbal command may be issued that contains the words “Good job. Stop.”. On its own, this single source of information is difficult to misinterpret. If, however, a strong expression of anger is visible at the same time as the spoken words, the underlying intent is more difficult to ascertain. The robot must determine if the person is in disagreement with the robot's interpretation of intent, and whether or not the action is being performed inappropriately. Perceiving a conflict between two or more modes of human expression is important to narrow down the underlying human intent and make the best use of perceived information while minimizing additional human effort.

5.1.2 Temporal Conflict

A human expression may not only be distributed across multiple modes, but also across time. Inconsistencies may arise in information as it is divided into multiple parts by the human mind to express over time. Similar inconsistencies may arise between concepts as they are interpreted by a robot using human expressions. Recently acquired information may be in conflict with prior hypotheses or information in human-robot interaction due to either inconsistencies introduced during human expression, or inconsistencies introduced during robot interpretation.

This disagreement between information at different points in time introduces significant challenges in human-robot interaction systems. Even if human expression is conveyed perfectly to the robot, the robot must distinguish between disagreements within the same phrase, and disagreements between successive phrases. A reasonable assumption can be made about the relationship between disagreements or inconsistencies between two pieces of information and the duration of time that separates them. As the time between two conflicting pieces of information increases, the significance of the inconsistency normally decreases. Inconsistencies within a single statement or phrase are based on concepts conveyed at similar times. An inconsistency within a single statement results in a self-contradicting statement, and is more significant than inconsistencies between concepts in the current statement and statements from long ago.

Temporal conflicts arise not only from inconsistencies between concepts over time, but also in inconsistencies between the action of a robot and external information including human intent. Conflicts between the action that a robot carries out and the intent of the human are, fortunately, straightforward to detect and resolve. When a robot carries out an action, the action is based on its understanding of human intent. Since the robot already believes that it is carrying out the desired intent, it has no means of detecting or resolving the conflict without additional information. Any conflict between action and intent must be handled by the human participant detecting and issuing a correction or conveying information about their disagreement. A temporal conflict can occur naturally and exist in the human expression, or may be introduced when a robot incorrectly recognizes or misunderstands the intent of human communication. Temporal conflicts present interesting challenges, since the significance of the conflict depends not only on the inconsistent information, but also the timing of that information.

A different form of conflict may arise not between elements of human expression, but between information in a robot's knowledge base. Development of a knowledge base is a feasible task that can be completed in reasonable time in highly constrained domains, but as the required scope of knowledge expands, it is increasingly difficult to ensure a knowledge base is consistent within itself. If all knowledge is represented in formal logic, consistency can be verified as each fact or relationship is added to the knowledge base. Since the knowledge representation used in this architecture is based on relatively independent cases, each case may be consistent with itself, but inconsistencies may still exist between two or more cases. Efficiently identifying and resolving inconsistencies in knowledge bases is an important area of research on its own [117, 118]. The limited scope of the interaction domain and manageable knowledge base size used in this research keeps the number of knowledge base inconsistencies to a minimum. Formal consistency checks should be performed when applying the presented framework in more complex interaction domains.

5.2 Detecting Conflicts

> On two occasions I have been asked by members of Parliament, *Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?* I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.
>
> — Charles Babbage

As Charles Babbage suggests, the correctness of a given solution relies on the correctness of its inputs. Unfortunately, in the presence of conflicts, the correctness of inputs cannot be guaranteed. Detecting the presence of a conflict or inconsistency provides information about the quality of the inputs, and can be valuable to avoid meaningless or incorrect solutions. Conflicts may be introduced everywhere from false human expressions or *inputs* that do not agree with the underlying human intent through to inconsistencies between multiple sources and even inconsistencies in the model itself. The first step in leveraging conflict to improve the quality of a robot's understanding of human intent is to acknowledge the existence of any inconsistencies, and to identify where and when a conflict may arise. The first source of conflict, mode conflict, requires knowledge of the sources from which the robot acquired possibly inconsistent information. In the presented framework for multimodal human-robot interaction, concepts are independent of any source, and therefore are of little use to directly detect mode conflicts. Following the data flow in reverse, back to lower levels, features are tied directly to a specific data acquisition source, and primitives are often associated with a single source.

Mode conflicts can be detected directly between low level features if they are related over short time durations and a reasonable measure can be defined to quantify inconsistencies. The correlation between features such as lip movements and audible speech provides valuable information in the form of constraints that can be used to detect inconsistencies [119]. Both the lip movement and audible portion must be consistent with the machine interpretation of generated speech. When closely correlated sources are available, the presence of conflict can be determined using any number of several measures of consistency with a carefully selected application specific threshold [120]. However, in the case of hand gestures and facial expressions, features are not as closely related over short time durations as with lip movements and speech. No single stream of information similar to speech is available as a model to define a reasonable measure of consistency between hand gestures and facial expressions. Inconsistencies between these two loosely-coupled sources of information about human intent should be detected using more than solely feature level information.

Once primitives are recognized, information about both the primitives and the possible concepts that they may form can be used to quantify inconsistencies between primitives. Since most primitives are based on a single modality when perceiving hand gestures and facial expressions, detecting a conflict between primitives will usually indicate a mode conflict. In Chapter 3, primitives are described as concurrent over all available sources of human expression, but sequential within a single channel. Only one primitive can be formed in each channel at any point in time.

Concept definitions can also be valuable to help identify inconsistencies, since each concept is derived from a combination of primitives over time. The explicit concurrency of primitives and concepts results in inevitable conflicts, but it is unreasonable to assume that all concepts and primitives are purely sequential. When two or more primitives in the same channel are good candidates for a given a sequence of observations, they are in conflict (a mode conflict). Only one of these primitives may exist at any point in time as they are recognized in the same channel. This conflict can be avoided by simply selecting the single most likely primitive in each channel at any point in time, but such a myopic approach focuses on the local channel without considering the big picture; the overall intent. The top n possible primitives in each channel are preserved to provide some flexibility when resolving conflicts between primitives in the same channel.

The information about how primitives and channels are related is also used together with concept definitions to identify mode conflicts. Inconsistencies between primitives are detected when combining sets of these concurrent primitives to form concepts. A predefined set of primitives are expected for any given concept. Each concept is associated with a set of relevant primitives using prior knowledge about the relationships between the concept and its morphology across available channels as described in Section 4.1.1. If a primitive from one channel suggests concept A, the remaining channels can be verified against concept A for consistency. A mode conflict exists when two or more of these concepts can be generated using the same set of channels but based on different primitives.

In the most trivial scenarios, all likely primitives can be broken into disjunct sets by channel, with a one-to-one mapping between a valid concept and each disjunct set. In this scenario, no mode conflicts are possible since no channel is shared between two or more concepts. Unfortunately, human expression can seldom be modelled in such a trivial manner. Overlap normally exists between sets of channels, resulting in different primitives from a single channel belonging to different concepts, as shown in Figure 5.2. It is also possible to generate multiple concepts from disjunct sets of primitives that have substantial overlap in their underlying observations. Since no constraints are imposed on the concurrency of concepts, primitives can be safely shared between concepts without directly introducing conflicts.

Not all conflicts can be identified solely using concepts and their constituent primitives. Some conflicts may arise in the context of communication, or knowledge about the surrounding concepts. A temporal conflict is detected when a concept cannot be incorporated into the analogy

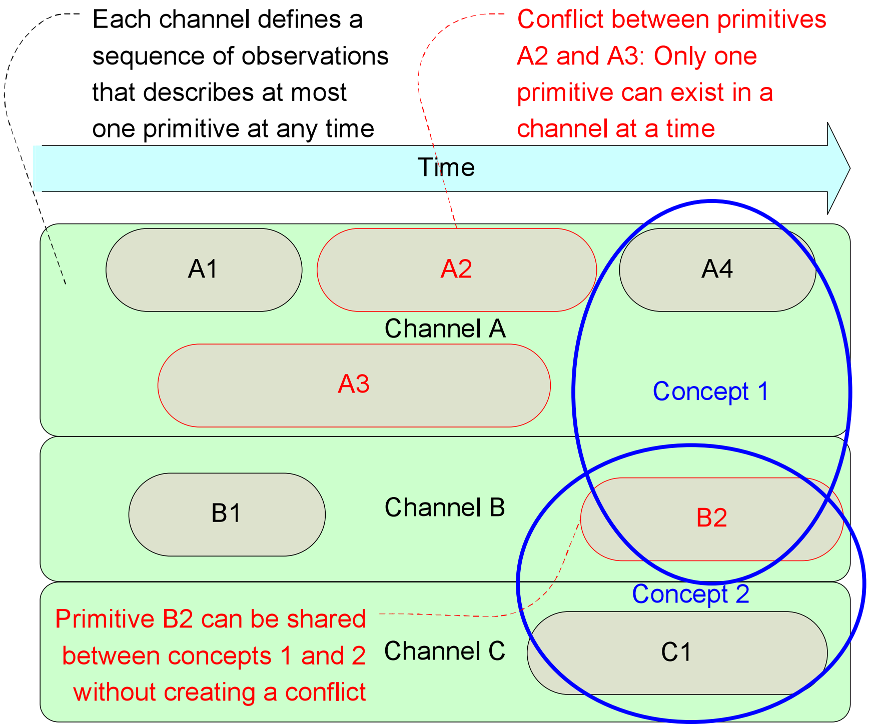

Each channel defines a sequence of observations that describes at most one primitive at any time
Conflict between primitives A2 and A3: Only one primitive can exist in a channel at a time
Time
A1
A2
A4
Channel A
A3
Concept 1
B1
Channel B
B2
Concept 2
Primitive B2 can be shared between concepts 1 and 2 without creating a conflict
Channel C
C1

Figure 5.2: Conflicts may exist between alternate primitives for the same observations, between two or more plausible concepts that share a common set of primitives, or between a concept and its surrounding context.

formed by the previous or succeeding sets of concepts. Although the n best concepts are maintained, the best concept as identified by a set of primitives may not be related to the few prior or succeeding concepts. Detecting a conflict of this type relies heavily on the accuracy of existing domain specific knowledge, and the quality of the preceding and succeeding concepts.

Detecting a temporal conflict is important, not only to improve the quality of the overall communication, but to help define a change from one analogy to the next. When a concept is unrelated to immediately preceding concepts, it may signify the start of a new phrase or set of concepts. Detecting a temporal conflict can help to segment phrases or analogies when segmentation based on lower level information is unsuccessful.

Once a reasonable analogy is found, the robot can attempt to carry out an appropriate action. In any real-world communication scenario, it is possible for the communicated intent to be misunderstood by one party without realizing. It is possible for an analogy to be found that provides a good explanation for the perceived primitives, but is not similar to the desired human intent. Whether the error was made during the human expression, during the recognition and understanding process, or embedded in the knowledge base, the robot cannot automatically detect this type of conflict.

A conflict between human intent and robot understanding may be detected only after the robot starts to carry out its action. Unlike mode and temporal conflicts, an action conflict is detected not by the robot, but by the human participant. When the human participant notices the discrepancy between their intent and the robot's action, they may communicate their concern. An action conflict like this is perhaps the most challenging to detect, as human feedback may be given to change or correct a robot's action for any number of reasons. The reason behind a command correction is opaque to the robot. The person may have simply changed her mind, or she may not have expressed what she thought she was expressing in the first place. Nevertheless, the end result should be the same when a correction is received, a change in action should occur. Since the end result is the same, our approach does not attempt to distinguish between an action conflict and a human change of mind.

5.3 Resolving Conflicts

Detecting a conflict can be useful to identify ranges of time in which perceived expressions are suspect, but defining a strict boundary between conflicting and non-conflicting is both difficult in the general case, and of little value to improve the quality of understood human expression. Defining a strict boundary is difficult in the presence of ambiguities, multiple interpretations, and even simple sensor noise. At this point, one must take a step back to reconsider the reason for leveraging conflicts.

Identifying the presence of conflicts is valuable as it provides a quantifiable measure that can be used to refine hypotheses of human intent. Selecting primitives and concepts to minimize conflicts is a reasonable objective toward improving the quality of the resulting hypothesis of human intent. Several options exist to minimize conflicts, many of which are specific to one source. In the case of conflicts between two directly competing primitives, one primitive can simply be chosen to override the other. A resolution of this type is considered *authoritative* if considered using a developmental model of conflict analysis [121], and can be useful if knowledge exists about the quality of information between channels. For example, if prior knowledge is available that gives a probability of false generation or perception from a given channel, it can be used in an authoritative approach to resolve a conflict. Using the prior example with a deictic hand gesture and a facial expression providing different spatial references, if it is known that a deictic hand gestures are less likely to be perceived incorrectly, the hand gesture information can be used to override the facial expression information.

Unfortunately, a naïve decision to select one overriding primary channel without additional information about the situation discards potentially valuable information conveyed by secondary channels or context, eliminating many of the advantages of multimodal communication. In addition, information about the consistency between any given channel and the underlying human intent is difficult to obtain and is often dependent on the intent itself. Information available across all channels must be considered together with the concepts that they may be expressing when resolving a conflict.

5.3.1 Resolving A Mode Conflict

The concept that provides the best explanation for a given set of conflicting primitives can be used as an authoritative source to resolve a mode conflict. However, a decision of this nature does nothing beyond what has already been accomplished during integration and understanding. The previous objective in integration and understanding was to select concepts to maximize similarity between perceived human expression and domain specific knowledge. A second objective is introduced when resolving a mode conflict, not based on similarity to existing knowledge, but based on inconsistencies between perceived information across different channels. Selecting a concept to minimize conflicts between primitives leverages multimodal information to resolve conflicts, but must be used in combination with similarity measures to provide meaningful results.

Since the difference between conflicting concepts may be small, and the presented approach relies on concurrency with multiple possible candidate primitives and concepts, the number of conflicts between primitives is often high and similar across many of the selected concepts. Simply minimizing the number of conflicting primitives is one approach, but provides little value when the number of conflicts does not vary substantially between solutions. Each concept (i) involved in a mode conflict covers a specific time span (t_i), and provides an explanation for a set of n primitives ($\{p_{i,0}, p_{i,1}, p_{i,2}, \cdots, p_{i,n}\}$). The quality of each concept, as previously obtained during integration, and the rank of each primitive is used to discriminate the level of conflict in concepts even though the number of conflicting primitives may be similar. The objective during mode conflict resolution is to select a subset of the plausible concepts to maximize the relevance of all selected concepts while minimizing gaps or periods of unexplained observations in any channel, and minimizing overlapping primitives in the same channel as shown in Figure 5.3. As previously suggested, this is not an optimization problem with a single optimal solution, but a multi-criteria optimization problem. The trade-offs between each of the these criteria must be considered.

The objective function shown in Equation 5.1 is used to evaluate each conflicting alternative. The trade-offs between concept relevance, unexplained observations, and overlapping primitives are captured in the parameters $w_{relevance}$, $w_{unexplained}$, and $w_{overlap}$. The balance between these criteria, and corresponding values of these weights is determined based on the duration of time considered since the beginning of the current hypothesis of intent. The time-varying values of these weights emphasizes divergence or exploration early in the process with a high $w_{unexplained}$ and low $w_{relevance}$, and convergence or exploitation later in the process with low $w_{unexplained}$ and high $w_{relevance}$. The objective function is initially biased toward concepts that make use of available primitives. As time progresses, the bias of the objective function shifts toward minimizing the unexplained primitives.

In Equation 5.1, the bounded set of concepts being considered from the integration module is C, and the bounded set of plausible primitives from the recognition module P_{recog}. P is the subset of primitives selected to maximize the objective function ($P \subseteq P_{recog}$). $r_{i,j,t}$ represents the relevance of primitive i with respect to concept j at time t. $q_{i,t}$ is a measure of the quality of primitive i at time t, as determined during recognition. If two primitives i and k are from the same channel, and overlap at time t, $q_{(i,t)}q_{(k,t)} > 0$. It should be noted that this overlap

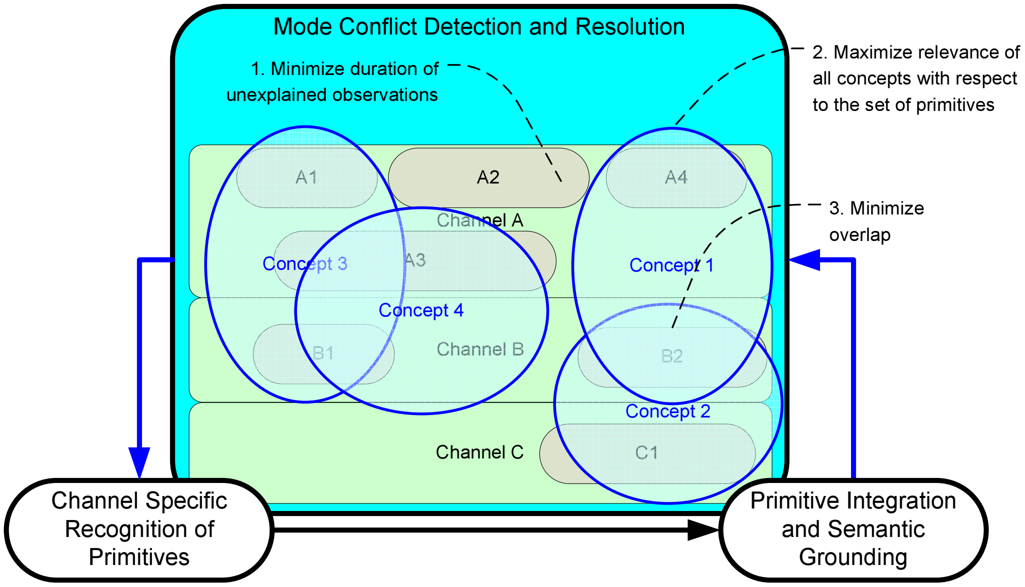

Mode Conflict Detection and Resolution
1. Minimize duration of unexplained observations
2. Maximize relevance of all concepts with respect to the set of primitives
3. Minimize overlap
A1
A2
A4
Channel A
A3
Concept 3
Concept 4
Concept 1
B1
Channel B
B2
Concept 2
Channel C
C1
Channel Specific Recognition of Primitives
Primitive Integration and Semantic Grounding

Figure 5.3: Mode conflicts are resolved by considering both the number and significance of conflicts between primitives when selecting relevant concepts to explain observed human communication over short durations of time.

includes both the scenario where one primitive ($i = k$) is included in multiple concepts in the set C, and where conflicting primitives from the same channel are included in the set of concepts C. $primitives_j$ represents the set of primitives included in the definition of concept j, $primitives_C$ represents the set of primitives included in the set of concepts C, and $overlap_C$ represents the set of pairs of overlapping primitives in the set of concepts C. $q_{i,t} = 0$ if primitive i does not exist at time t.

$$\begin{aligned} f_m(C,P) = & w_{relevance} \sum_{j \in C} \sum_{i \in (P \cup P_{recog}) \cap primitives_j} \sum_t (r_{i,j,t} + q_{i,t}) \\ & - w_{unexplained} \sum_{i \in P \setminus primitives_C} \sum_t q_{i,t} \\ & - w_{overlap} \sum_{(i,k) \in overlap_C} \sum_t q_{(i,t)} q_{(k,t)} \end{aligned} \tag{5.1}$$

The optimal set of primitives, P, is selected from both possible primitives included in each concept definition, and primitives provided by the recognition module. The optimal set of primitives, P, is used to provide feedback to the recognition module. Primitives that will help minimize mode conflicts, $P_{recog} \setminus (P \cap P_{recog})$, are identified as desirable by issuing one positive vote for each of these primitives, whereas primitives that cause mode conflicts, $P \setminus (P \cap P_{recog})$, are issued one negative vote to influence the preceding ranking and selection process in an attempt to resolve mode conflicts. Votes accumulate for each primitive instance over the duration of the current analogy. This information is provided in normalized form ($-1 \leq V_i \leq 1$) as feedback to each primitive instance i.

It is important to note that the iterative feedback process and bounded set of primitives and concepts is used to both limit evaluation complexity, and to keep communication across well-defined boundaries between modules restricted to the necessary information. Exposing all primitives and concepts, including the highly unlikely ones in this conflict resolution process will increase the time required to find the set P. Using a bounded set of pre-screened primitives and concepts from the recognition and integration modules is not perfect, but it eliminates the need to evaluate primitives and concepts that are extremely unlikely to contribute to the resulting understanding of human intent. The accumulation of votes and feedback process provides information to refine the set of primitives and concepts if necessary while maintaining the same bounds. The votes are used in the recognition and integration modules to re-evaluate the ranking of primitives and concepts over a given time frame. Primitives that may otherwise be discounted and not passed on to the integration module may, after incorporating feedback from conflict resolution, be included in the output set of primitives.

Each positive vote can increase the rank of a primitive relative to alternatives from the same channel. Similarly, each negative vote can decrease the relative rank of a primitive. Each primitive λ_i in each channel for in a given time frame is ordered using $\alpha_p \hat{P}(O|\hat{\lambda_i}) + (1 - \alpha_p)V_i$, where V_i is a normalized measure of votes between -1 and 1. V_i is provided as feedback for primitive i, and α_p is a tuning parameter to balance the rate of influence of higher-level feedback against direct low-level observations. This low-complexity approach allows a bounded maximum to be imposed on the most appropriate primitives to be selected and passed along in a timely fashion. Re-ordering primitives minimizes tuning requirements, while the upper bound on appropriate primitives limits the size of the search space in subsequent modules. The resulting high-ranking primitives may not have the highest measures of possibility based solely on observations, but are ideally the most likely candidates when considering both observations and feedback from integration and analysis of conflict.

5.3.2 Resolving A Temporal Conflict

Unlike resolving a mode conflict directly between primitives, resolving a temporal conflict requires historical information about previously identified concepts. A concept may be inconsistent with other concepts in close proximity, either older or newer. Since the resolution of a temporal inconsistency relies on information about surrounding concepts, the most recent concepts only have a portion of the information required to determine their involvement in a conflict. For example, if an attempt is made to express the intent to *put the dish on the table, answer the door*, and the concepts *put* and *table* are currently identified by the integration module, insufficient information exists to resolve a temporal conflict. These can be resolved only after identifying the following concepts and forming hypothesis of intent.

Temporal conflicts can be resolved once time elapses, producing either succeeding concepts or a significant break in communication. When succeeding concepts are available, information about

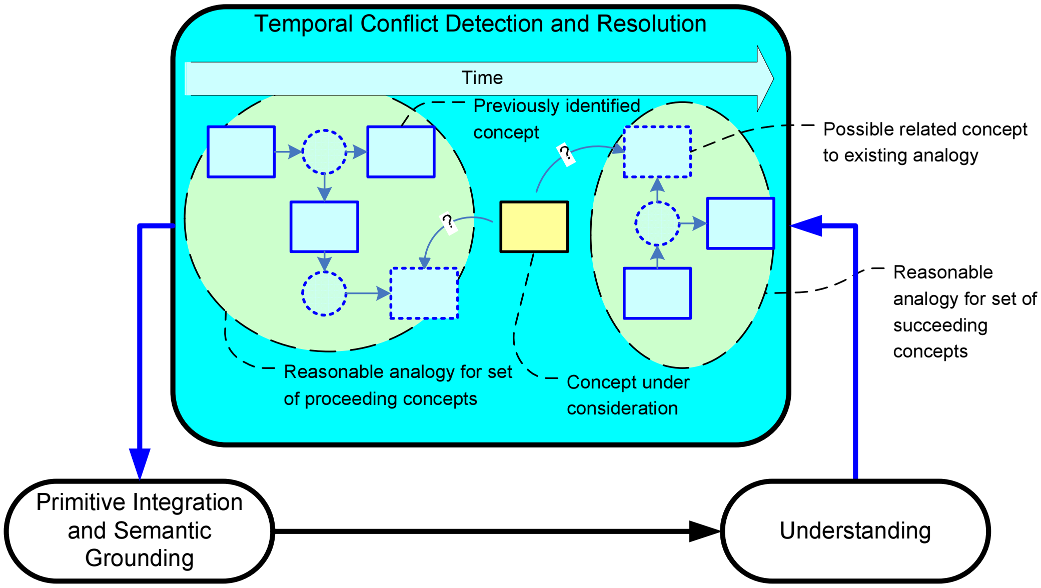

Temporal Conflict Detection and Resolution
Time
Previously identified concept
Possible related concept to existing analogy
?
?
Reasonable analogy for set of succeeding concepts
Reasonable analogy for set of proceeding concepts
Concept under consideration
Primitive Integration and Semantic Grounding
Understanding

Figure 5.4: Temporal conflicts may be resolved by considering the context surrounding a concept, and the relative importance of incomplete analogies in terms of time and channel coverage.

the analogies provided by the understanding module is used as context to identify and resolve any temporal conflicts. Any concept not belonging to the same analogy as its closest neighbours in time introduces a temporal conflict. Similarly, concepts belonging to an analogy missing essential concepts (as defined in the existing domain knowledge) and its neighbours may be in conflict. If the conflicting concept together with its succeeding concepts result in an adequate analogy, the concept is not considered to be involved in a conflict. However, if a concept does not belong in the same set with either its preceding or succeeding concepts, steps are required to resolve the conflict. An overview of these temporal conflict resolution issues are shown in Figure 5.4.

Several options exist to resolve a temporal conflict, including eliminating the concepts involved in a conflict, modifying the conflicting concepts to match surrounding context, or soliciting human assistance. If the preceding or succeeding sets of concepts are self-sufficient, where reasonable analogies can be found without the conflicting concept, a single conflicting concept may be discarded. Unfortunately, discarding a concept is not always desirable, especially when the concept can clearly explain a set of observed primitives.

A *P-mode* or power-struggle approach from developmental conflict analysis is applied to resolve temporal conflicts. This approach is described by a struggle for identity formation combined with competition for autonomy [121]. The participants in the struggle are the alternate analogies that can be formed from the available concepts. The space formed by the set of channels and time explained by each analogy is used to indicate the strength of each analogy. An incomplete analogy (including the case of an isolated concept) will not survive the power struggle when the space of time and input channels that it explains is small relative to the space of time and input channels explained by nearby sets of concepts.

If an incomplete analogy looses the power struggle, an attempt can be made to combine its concepts with surrounding analogies. If the preceding or succeeding analogies can incorporate the identified incomplete analogy while discarding a minority of the channel-time space described by its concepts, the incomplete analogy is combined with the more powerful nearby analogy. In cases where a majority of the channel-time space described by the incomplete analogy must be discarded in this combining process, an alternate approach must be considered.

When an incomplete analogy cannot be combined with a more powerful neighbour, one of two actions may be pursued. In the case where the incomplete analogy explains observations that all terminate older than $t - t_{adelay}$, it can be assumed that the analogy should have been completed before time t. Since the analogy is still incomplete, every concept in this analogy can be given negative feedback to encourage exploration of alternate concepts that may result in a more complete analogy that is a better fit within its context. Although this form of segmentation at the analogy level is beyond the scope of this research, valuable feedback about anticipated concepts is provided to improve the accuracy of identifying the current analogy. When an incomplete analogy exists with at least one concept explaining an observation in the interval $(t - t_{adelay}, t)$, it is likely that future observations will change and possibly complete the analogy. In this case, all absent but critical concepts in the existing analogy are given a positive vote when adjacent to or overlapping an existing concept match in the analogy (see Chapter 4 for further details).

Similar to how the feedback provided by analyzing mode conflicts is used to refine primitives, the feedback provided by analyzing temporal conflicts helps to refine concepts. Feedback provides hints in the form of a accumulated votes for concepts over a fixed time intervals. This feedback is iteratively incorporated into the inference process to refine the estimate of appropriate concepts. Feedback is provided in the form of positive votes for concepts in a given time frame that are in agreement with surrounding context, and negative votes for concepts causing the most conflict. In a similar approach to feedback used to improve recognition, these votes are used to adjust the relative rank of each concept in a given time frame.

Concepts in a given time frame are ordered by $\alpha_c c_i + (1 - \alpha_c)V_i$, where V_i is a normalized measure of votes for concept i given as a value between -1 and 1, and α_c is a tuning parameter to balance the rate of influence of feedback against the information provided based on direct observation. Larger values of α_c increase the influence of integration, and smaller values increase the influence of the feedback from temporal conflicts. The reason for using a cumulative voting scheme in this manner is the same as the one used in mode conflict; to limit the size of the search space, exploring further on an exceptional basis. One may consider this feedback approach a form of iterative deepening search, where the search terminates if a solution is found with no conflicts but continues deeper in an attempt to resolve any existing conflicts. The bounded set of analogies and concepts provided from the integration and understanding modules reduces the need to evaluate conflicts between analogies and concepts that are unlikely to persist until an action is performed.

The following chapter includes experiments to suggest values for the tuning parameters mentioned in the discussion of conflict analysis. These experiments analyze the performance of the presented architecture from perception through to understanding and conflict analysis, and provide concrete examples that may help to illustrate several of the approaches presented throughout the current and prior chapters.

Chapter Six

Implementation and Experiments

I hear and I forget.
I see and I believe.
I do and I understand.
— Confucius

The theoretical design of each subsystem in the multimodal architecture was discussed in Chapter 3 through Chapter 5, however, a physical implementation is valuable to demonstrate the approach in a tangible manner. From the perspective of an end user, a purely applied vantage point, the acceptance of any human-device interface depends on its real-world performance. This chapter shifts the focus from theory to practice to demonstrate an implementation along with experiments and tangible results. To help make this transition, the implementation of the previously discussed multimodal interaction architecture is presented using domestic service robot interaction scenarios. The service robot application is particularly relevant due to the current rapid growth and demand for increasingly intelligent devices to assist in domestic environments. Although currently mass produced domestic service robots are limited in scope, including vacuum cleaners and lawn mowers, an increase in the range of tasks is inevitable. Service robots must be flexible to effectively provide assistance in dynamic domestic environments and help the increasing elderly demographic. The discussed multimodal architecture provides an effective means to interact with these increasingly complex domestic service robots.

In addition to demonstrating an implementation of the architecture with a domestic service robot application, implementation challenges are also presented. The various obstacles encountered are included in this chapter, along with approaches used to overcome an obstacle when relevant. This chapter includes an analysis of the practical performance of each component in the system both as individual subsystems and together as a complete interaction system. For consistency with the previous chapters, the discussion and analysis of performance follows the flow of information along a logical path from human expression through to feature extraction, recognition of primitives, and understanding of intent.

6.1 System Architecture

A high level diagram of the implemented system is illustrated in Figure 6.1. The implementation is in C/C++, developed in a Linux environment, with cross-platform Windows/Linux support for most components. The cross-platform capabilities are provided by using the GNU CommonC++ library for communication and threading support, and the FOX toolkit for GUI support

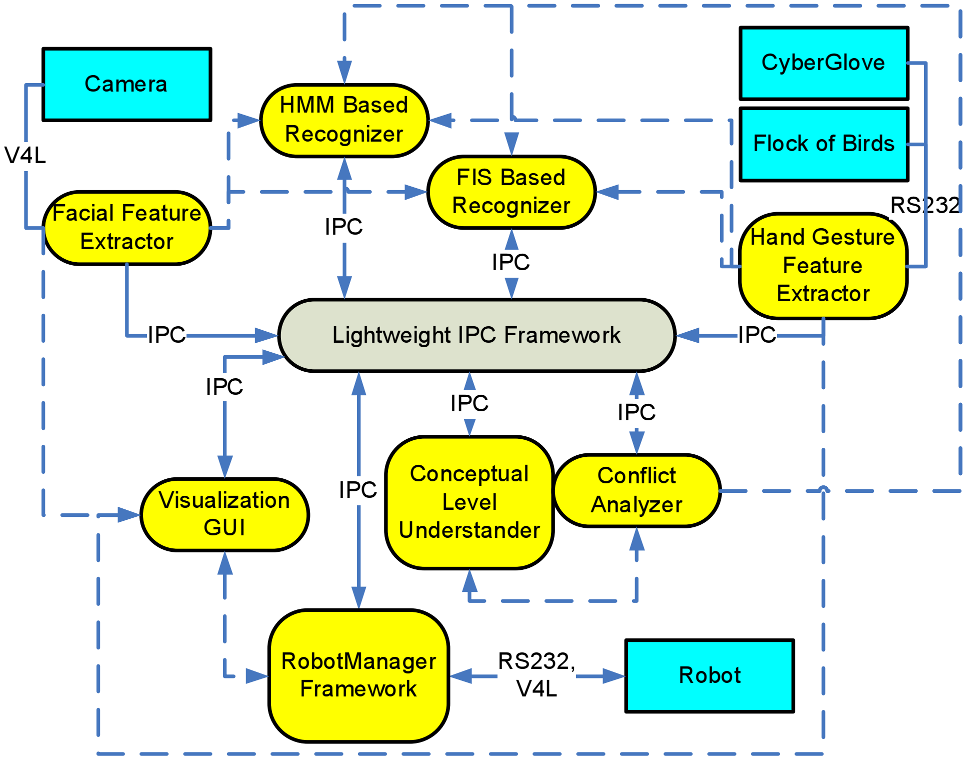


Figure 6.1: High Level Diagram of Implementation. Solid arrows indicate direct flow of information (primarily IPC), whereas dashed arrows indicate indirect flow of information between components.

As illustrated in Figure 6.1, two modality specific feature extraction components are implemented: one to extract features using the camera, and one to extract features using the Flock of Birds and CyberGlove. Communication between components relies on a lightweight inter-process communication framework. This framework was developed to provide simple but robust publish/subscribe channel based communication over unreliable connections, as experienced in many wireless infrastructures for mobile robots. The two feature extraction components publish both raw and feature information for any interested external application to use. Solid arrows indicate direct (real) information flow, whereas dashed arrows indicate indirect interactions between applications.

The two forms of primitives are recognized by two separate applications, the HMM based approach for dynamic primitives, and the FIS based approach for static primitives. These two applications subscribe to feature information relevant to each primitive in each independent channel, in addition to a primitive level feedback channel. Information about the top N recognized primitives is then published using IPC. All conceptual graph related processing is implemented in another application, which subscribes to primitive channels and a concept feedback channel. This application implements both the semantic grounding and the Tabu search algorithms.

An independent visualization interface is used to monitor the state of the system, and to provide qualitative visual feedback during operation. Similarly, an independent framework is implemented for low-level robot control. Both the visualization interface and the robot control framework take advantage of the existing lightweight inter-process communication framework to exchange control, monitoring, and perception information. Further details about the robot control framework and IPC approach can be found in Section 6.6.2 and Section 6.6.3.

6.2 Calibration and Data Acquisition

Before any device can understand human expression, it must first sense or otherwise receive information about the human generating the expression. Since facial features and hand gestures are the two selected forms of human expression in this application, sensors must be evaluated and calibrated to ensure the necessary information is acquired to perceive these forms of human expression. Hand gesture information is acquired using an electromagnetic tracker attached to the forearm together with a glove embedded with resistive bend sensors. This sensor data is transmitted over two dedicated serial connections to a computer, eliminating communication congestion and unpredictable latency issues that sometimes exist on shared media. Similarly, facial expression information is acquired using a dedicated camera and transmitted over a dedicated connection.

Any sensor used to provide information about human expression should be calibrated and characterized to provide measures of accuracy and precision valuable to form a sensor model. A vision based sensor may be incapable of discerning individual facial features, preventing the application of a local feature based facial expression approach. Similarly, a sensor that does not measure abduction between fingers, or provides joint angles using only a 3-bit encoding will present challenges distinguishing between spread and adjacent fingers or identifying more complex configurations, including contact between the thumb and fingertip.

6.2.1 Noise in Sensors

An accurate representation of the sources and characteristics of noise is important in any experiment involving physical or real world measurements. Sources of noise include equipment characteristics such as quantization errors and calibration, external sources including electromagnetic disturbances in the hand tracking system and light conditions in the vision system, and human sources including variations in timing and position of expressions. Although an additive white noise model is commonly used to characterize noise, it is important to first measure this noise to justify any selected model. With knowledge about the underlying distribution of the noise model, future predictions can be made about how noise will impact the sensed data.

Determining the precision of the information acquired about the hand and face serves two purposes; to help build sensor models, and to identify differences between the current implementation and alternate approaches to perceive human expression to simplify interchanging sensors in future work. As previously noted, a glove-based implementation is helpful in the scope of the current research, but a transition from a glove-based sensor to a more transparent sensor will broaden the applications in which this architecture can be successfully applied. Stationary accuracy and precision of the hand tracker was quantified using a set of known points in the volume of hand movement required for interaction. Known points were labelled on a surface which was positioned at known distances and orientations from the transmitter, at 100mm, 200mm, 300mm, 400mm, and 500mm distances. The receiver was placed on each of the calibration points for sufficient time to collect between 1000 and 5000 samples (10-50 seconds @ 100Hz). To reduce the influence of systematic errors, including variations that occur as the temperature of the equipment varies, the calibration process was repeated 3 times over the duration of one hour.

After collecting samples from the hand tracker in each stationary position, results were plotted in a histogram to visually observe the distribution of the sensor noise. The typical symmetric bell-shaped curve is visible in Figure 6.2, suggesting a normal distribution. To confirm this hypothesis using a formal method, a Shapiro-Wilk W test for normality was performed. This test is a common statistical procedure to measure the distance between a small to medium set of samples (≤ 5000) and a normal distribution. The measure W is similar to a correlation coefficient between the squared ordered sample values and normal scores as shown in Equation 6.1 [122]:

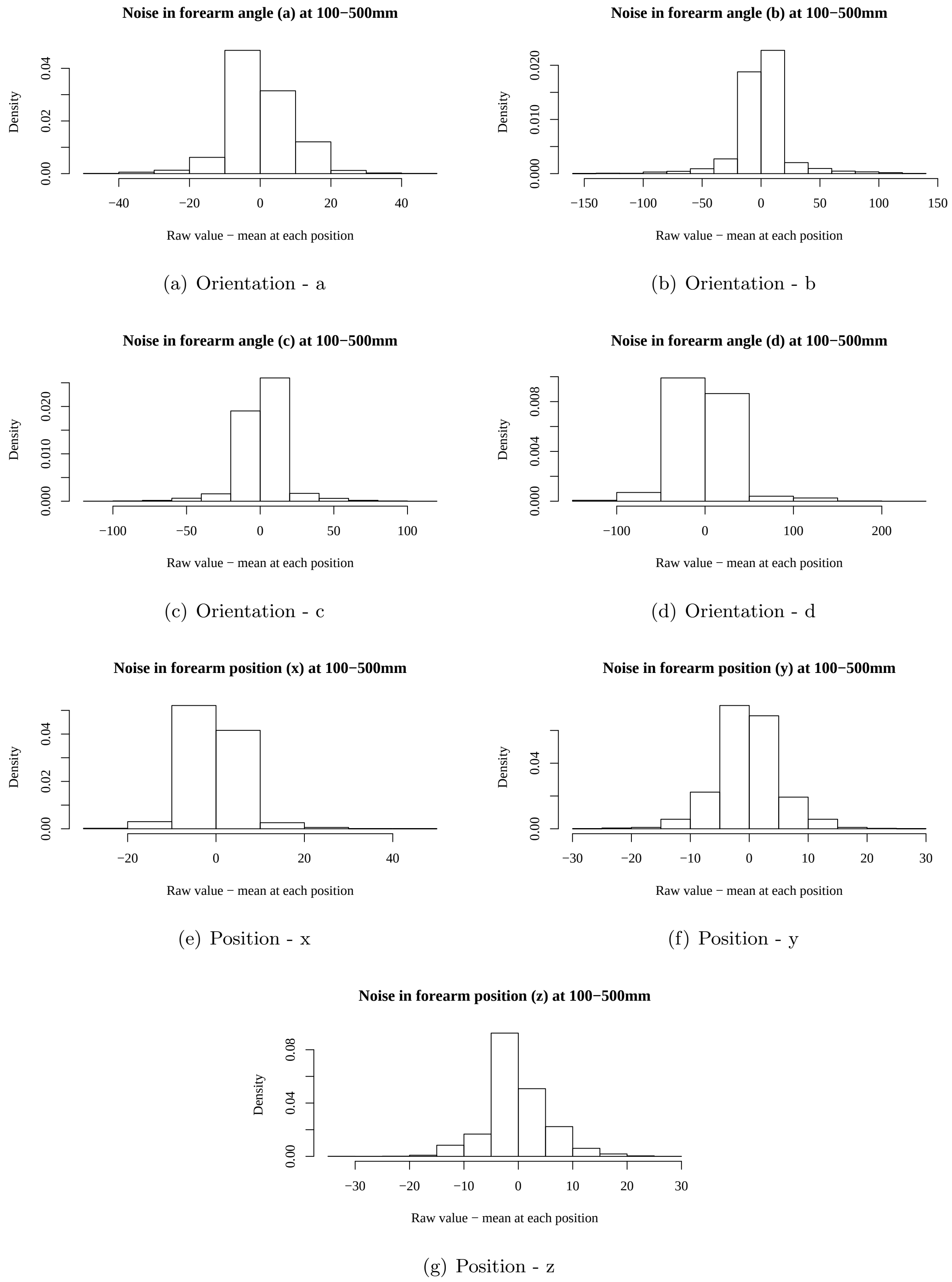
$$W = \frac{\left(\sum_{i=1}^{n} w_i x'_i\right)^2}{\sum_{i=1}^{n} \left(x_i - \bar{x}\right)^2}. \tag{6.1}$$

Where, $W = 1$ when the samples are normally distributed, n is the number of samples, x is the original sequence of samples, x' is the sequence of samples ordered in non-decreasing order (i.e. x_i is the i^{th} smallest value), and w_i is a weight similar to the normal distribution. In the Shapiro-Wilk test, the weights $w_1, w_2, \cdots, w_n$ are calculated as,

$$w_1, w_2, \cdots, w_n = \frac{\mathcal{M}^T \mathcal{V}^{-1}}{(\mathcal{M}^T \mathcal{V}^{-1})(\mathcal{V}^{-1} \mathcal{M})}, \tag{6.2}$$

where $\mathcal{M}$ is a vector containing the expected values for a standard normal distribution over sample size n, and $\mathcal{V}$ is the covariance matrix [123].

From direct observation of the distribution of noise shown in the histograms in Figure 6.2, stationary noise for hand position and orientation measurements is focused tightly around the mean, and, at first glance, appears similar to the typical bell-shaped normal curve. The p-values of the Shapiro-Wilk test for normality are consistently smaller than 0.001, forcing the hypothesis of normality to be discarded, or a normal distribution is not a good fit for the noise. If one inspects the distribution of noise in the quantile-quantile (Q-Q) plots shown in Figure 6.3, it can be seen that the distribution of noise is heavy-tailed, causing the failed Shapiro-Wilk test for normality. A normally distributed set of data points should lie close to the line shown in these plots. Even though the distribution is heavy-tailed, it is extremely tight around the mean relative to the full scale of values. Each value is 16-bits, with a full range of -32768 to 32767. All measured angular values were within 150 units of the mean, or less than 0.45% of the full range. Similarly, all measured position values are within 30 units of the mean (omitting the outliers in Figure 6.3(e)), or less than 0.1% of the full range (914mm). Angular errors of less than 0.45% and position errors less than 1mm are extremely small in the context of human expression. It is reasonable to assume this form of noise can be considered negligible relative to the noise generated from a human hand.



(a) Orientation - a

(b) Orientation - b

(c) Orientation - c

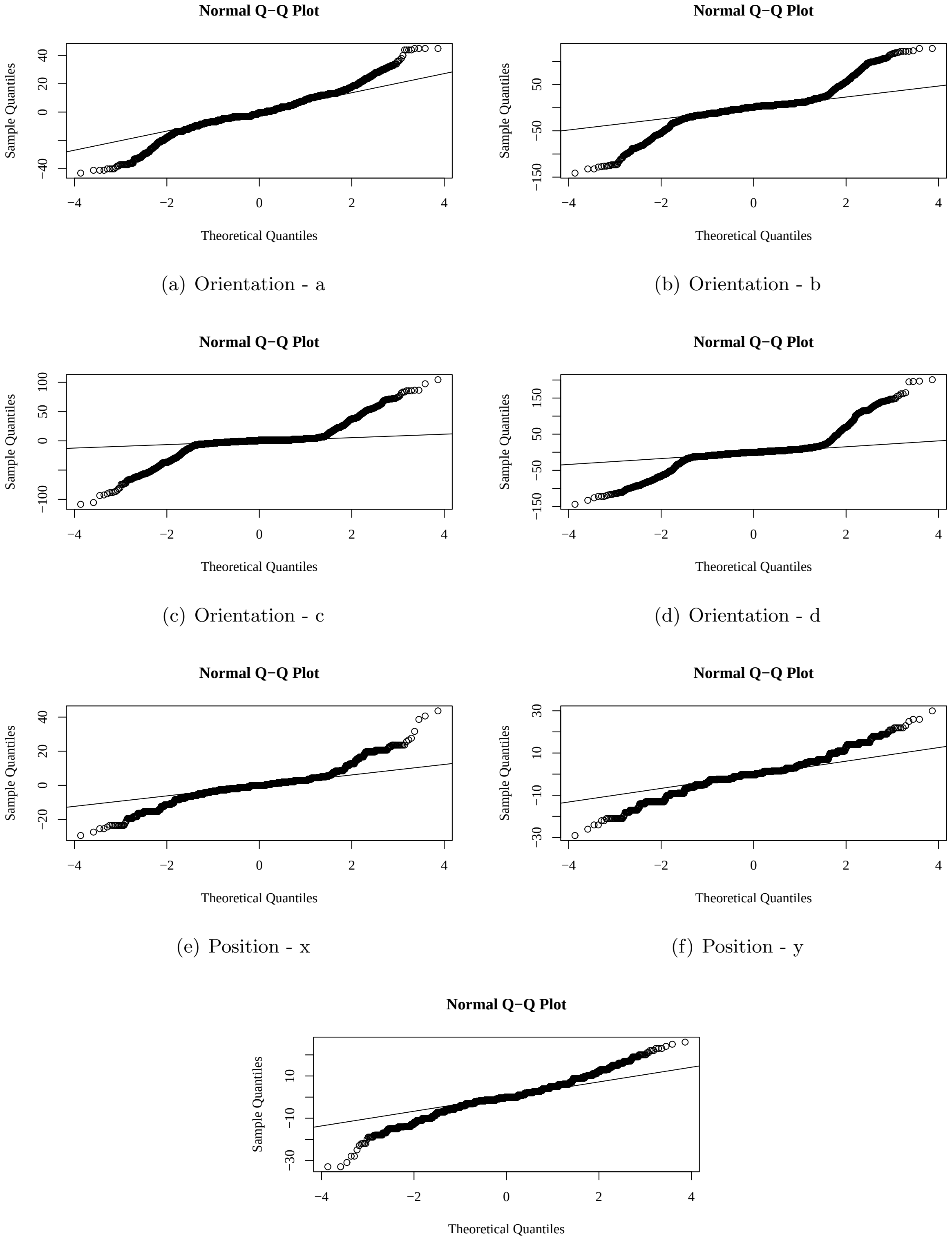
(d) Orientation - d

(e) Position - x

(f) Position - y

(g) Position - z

Figure 6.2: Distribution of Noise in Hand Position and Orientation Measurements



(a) Orientation - a

(b) Orientation - b

(c) Orientation - c

(d) Orientation - d

(e) Position - x

(f) Position - y

(g) Position - z

Figure 6.3: Q-Q Plots of Noise in Hand Position and Orientation Measurements

Table 6.1: Values for the user-dependent calibration parameters for each joint angle are determined by comparing measured raw values against known hand structure limitations with visual verification.

Joint	Description	scaling: α_i	offset: b_i	$\mu - 2s$	$\mu + 2s$
0	Thumb CMC (basal) flexion (radial abduction)	0.0159	21	-32	44
1	Thumb MCP joint flexion	0.0145	97	-49	51
2	Thumb IP joint flexion	0.0143	109	-51	46
3	Thumb-Index CMC (palmar) abduction	0.00505	81	-88	123
4	Index MCP joint flexion	0.0162	70	-65	44
5	Index PIP joint flexion	0.0146	77	-59	51
8	Middle MCP joint flexion	0.0173	63	-87	33
9	Middle PIP joint flexion	0.0180	80	-65	50
11	Middle-Index MCP joint abduction	0.00450	151	-89	82
12	Ring MCP joint flexion	0.0176	75	-71	34
13	Ring PIP joint flexion	0.0179	91	-63	62
15	Ring-Middle MCP joint abduction	0.00616	162	-71	67
16	Pinkie MCP joint flexion	0.0166	85	-91	65
17	Pinkie PIP joint flexion	0.0182	80	-96	45
19	Pinkie-Ring MCP joint abduction	0.00600	131	-76	67
20	Palm Arch	0.00912	71	-89	63
21	Wrist Pitch	0.00869	143	-98	66
22	Wrist Yaw	0.00394	89	-118	67

The resistive bend sensors used to measure hand configuration exhibit no noise above the level of quantization (8-bits for each of the 18 sensors). Similarly, no discernible noise exists in the vision system above the level of quantization. Unfortunately, the precision of a sensor does not necessarily correlate with accuracy. The accuracy of each sensor should also be determined to properly calibrate raw values against real-world measurements.

As previously shown, variations in the position and orientation of the hand tracker are negligible. The accuracy of forearm position is used to calibrate the tracker, and to help locate any regions influenced differently by external sources of interference. Deviations from equipment readings and real-world measurements are compensated using an offset, scaling factor, and a usable linear region between 150mm and 750mm from the transmitter. A similar calibration exercise was performed to determine the accuracy of the resistive bend sensors measuring the configuration of the hand (summarized in Table 6.1, Figure 6.5, and in Chapter 3 where the sensor was introduced). Finally, the vision based component to acquire facial expression information was used to measure manually labelled key points on a near-frontal face at a fixed distance from the camera. Examples of these key points can be observed in Figure 6.4. At a distance of approximately 600mm, the neutral expression and near extreme values for each key point on the face are summarized in Table 6.2.

This analysis of data acquisition equipment focused on sensor noise to determine the suitability of each sensor for measuring human expressions. Although a minor level of noise exists in the position and angular measurements of the forearm, the level of noise is well below reasonable expectations of variations in forearm movements. Quantization error is a more significant influence

Table 6.2: Values for the user-dependent calibration parameters for each facial feature are summarized in this table. All values are normalized to the face region with a centre origin (0,0) for consistency at varying scales.

	Keypoint	neutral (x,y)	$\mu - 2s$ (x,y)	$\mu + 2s$ (x,y)
Lips	Left endpoint	(-0.40,0.55)	(-0.45,0.46)	(-0.14,0.77)
	Right endpoint	(0.21,0.63)	(0.17,0.51)	(0.35,0.85)
	Centre of upper lip	(-0.09,0.51)	(-0.12,0.48)	(0.07,0.67)
	Centre of lower lip	(-0.09,0.65)	(-0.12,0.51)	(0.08,0.92)
Eyebrows	Left endpoint	(-0.50,-0.43)	(-0.62,-0.61)	(-0.35,-0.26)
	Right endpoint	(0.60,-0.23)	(0.62,-0.33)	(0.73,-0.18)
	Mid-left peak	(-0.20,-0.50)	(-0.55,-0.68)	(0.06,-0.39)
	Mid-right peak	(0.33, -0.45)	(0.19,-0.55)	(0.44,-0.40)
	Centre	(0.04,-0.29)	(-0.09,-0.48)	(0.12,-0.26)
Eyes	Left	(-0.26,-0.23)	(-0.48,-0.37)	(-0.06,-0.14)
	Right	(0.31,-0.13)	(0.23,-0.22)	(0.49,-0.04)

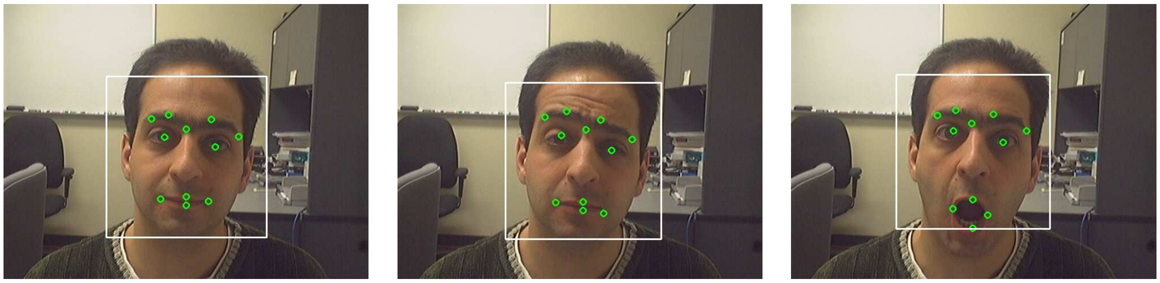


Figure 6.4: A set of keypoints are identified and tracked using user-dependent multidimensional histograms and geometric constraints to quantify information about human facial expressions.

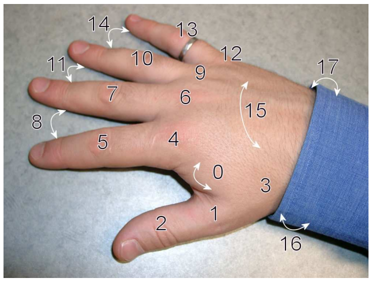


Figure 6.5: A set of hand joints are measured using resistive bend sensors to provide information about hand gesture components of human expressions.

than sensor induced noise or external noise for the resistive bend sensors in the glove and CCD sensors in the camera.

6.2.2 Noise in Human Expressions

On their own, the accuracy and precision of the position, orientation, and configuration of the resistive bend sensors and camera does not explain how this equipment impacts machine perception and, ultimately, the performance of the overall system. Variations in human expression must also be characterized before commenting on the suitability and impact of the sensors and data acquisition methods. Basic characteristics of noise in the generation of human expressions were determined by sampling human hand and facial feature movements. It is expected that variations across even stationary human expressions are significantly larger than the precision of the sensors, but this hypothesis must be quickly tested before proceeding. The results presented in Figure 6.6, and Figure 6.8 are based on holding one pose as stationary as possible without supplemental arm or head support for 30-45 seconds. Variations in resistive bend sensor values and pixel positions were distributed tightly around the mean, with the exception of a few rare outliers. As confirmed by this quick experiment, variations across repeated human expressions have a larger impact than previously measured static noise in an isolated sensor. It should also be noted that the distribution of human-generated forearm position noise is a closer fit to a normal distribution than the previously measured stationary sensor noise. This can be qualitatively observed in the Q-Q plots for selected face keypoints in Figure 6.9, and in the Q-Q plots for hand position in Figure 6.7 when compared against the equivalent plots in Figure 6.3.

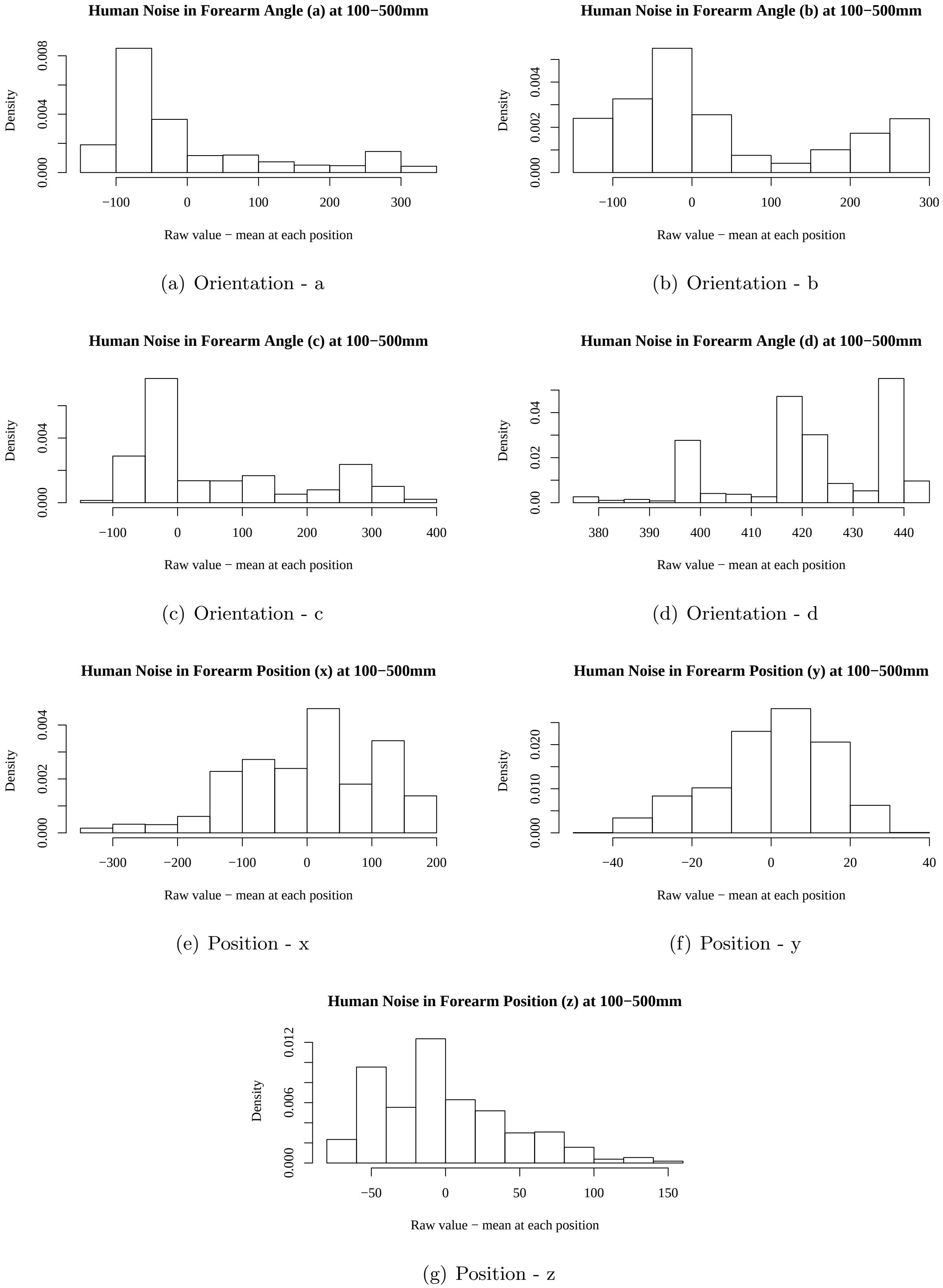
6.3 Training and Recognizing Primitives

As described in Chapter 3, the recognition process focuses on recognizing a small set of concurrent primitives. It is important to measure the performance of this recognition process for individual primitives before moving ahead to concepts and hypotheses of intent. Since primitives are the building blocks for all higher level processing, accurate recognition of primitives is desirable. Perfect recognition cannot be expected, but unchecked inaccuracies in the recognition of primitives will cascade through to impair the performance of subsequent tasks, reducing the quality of concepts and any resulting hypotheses of intent.

The two distinct forms of primitives, static and dynamic, are evaluated independently for each source of information about human expression. The static and dynamic forms are isolated for numerous reasons, including differences in noise characteristics, and differences in the algorithms used to recognize each form. Static primitives are modelled a fuzzy inference system, as described in Chapter 3, whereas dynamic primitives are modelled using HMM variants.

6.3.1 Hand Gesture Primitives

It is logical to present the performance of each hand gesture primitive in relation to the features it requires, and in relation to the possible concepts that it may help express. A typical example of



(a) Orientation - a

(b) Orientation - b

(c) Orientation - c

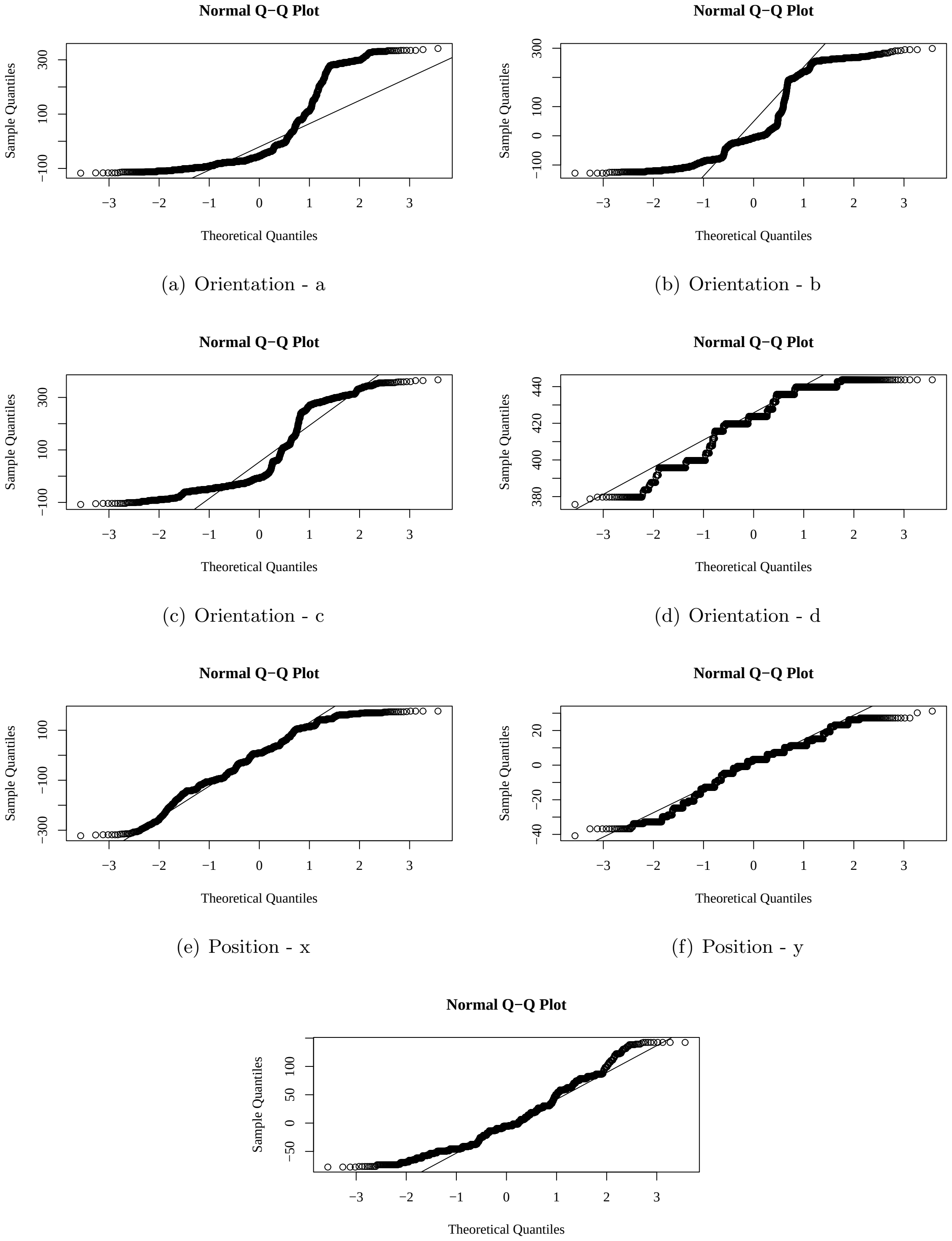
(d) Orientation - d

(e) Position - x

(f) Position - y

(g) Position - z

Figure 6.6: Distribution of Human-Generated Noise in Hand Position and Orientation Measurements



(a) Orientation - a

(b) Orientation - b

(c) Orientation - c

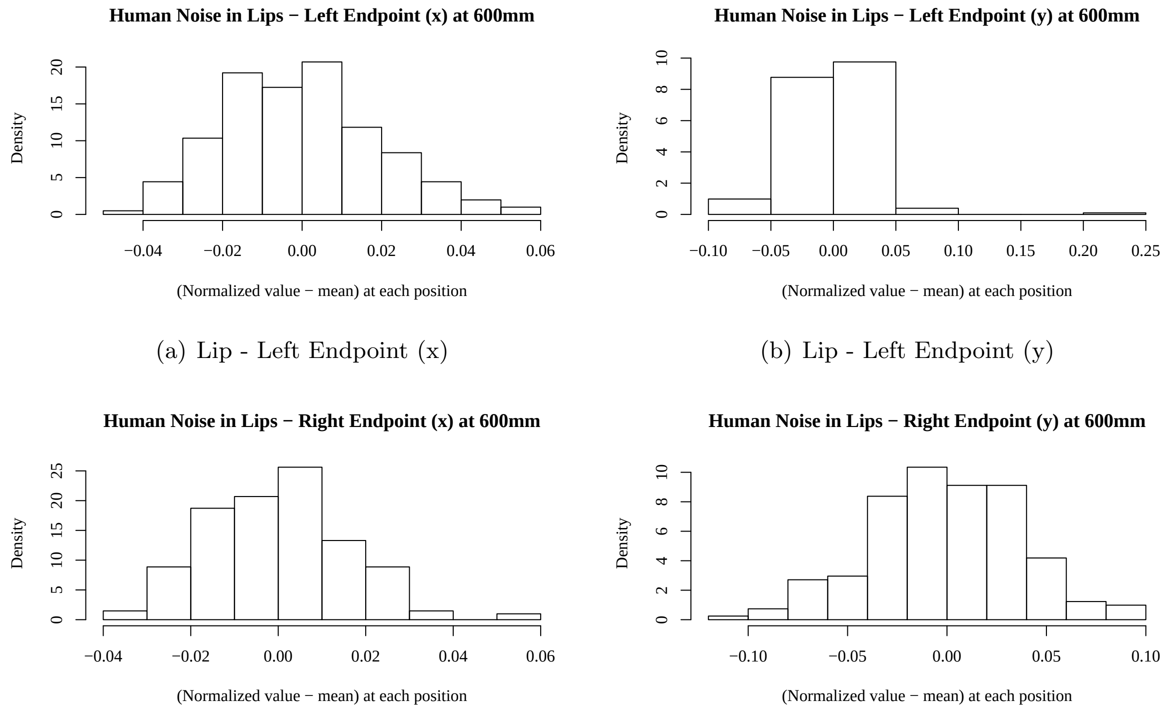
(d) Orientation - d

(e) Position - x

(f) Position - y

(g) Position - z

Figure 6.7: Q-Q Plots of Human-Generated Noise in Hand Position and Orientation Measurements



(a) Lip - Left Endpoint (x)

(b) Lip - Left Endpoint (y)

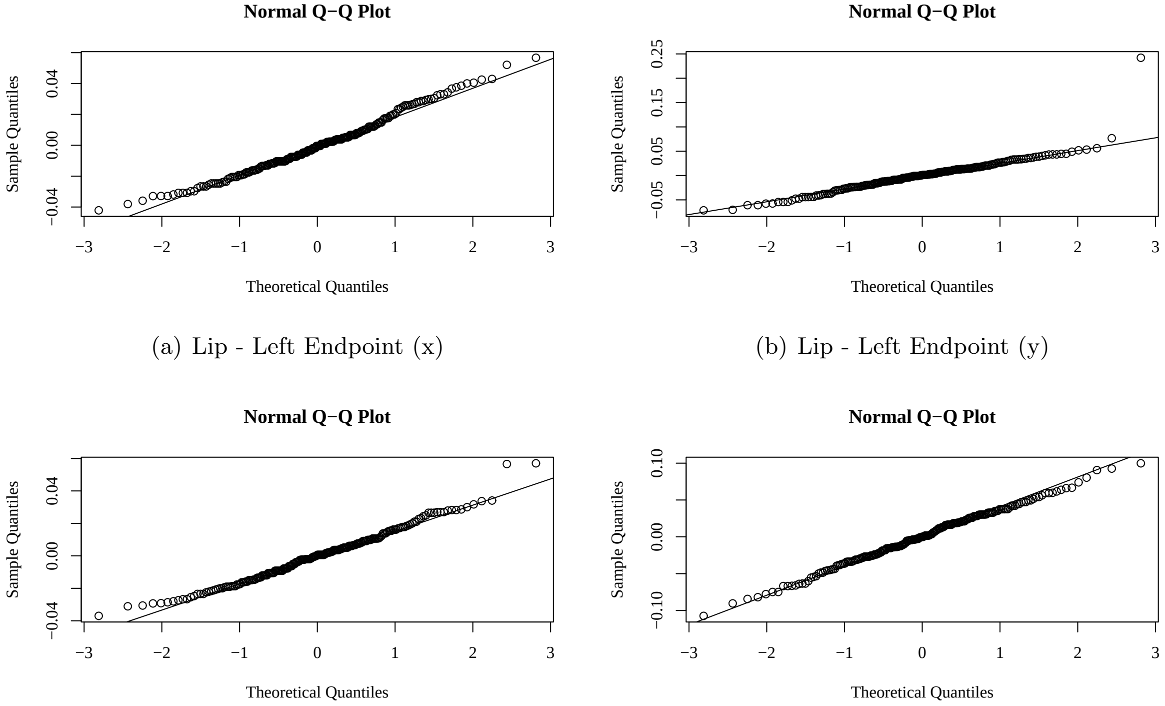
(c) Lip - Right Endpoint (x)

(d) Lip - Right Endpoint (y)

Figure 6.8: Distribution of Human-Generated Noise in Selected Face Keypoint Measurements

one of these primitives is the flat hand, where each finger is extended to align in the same plane as the palm. The flat hand primitive is modelled as a static primitive with no time component, but it is also valuable to model the dynamic motion from a more natural or relaxed hand configuration to the flat hand (i.e. extending fingers). Features required for the flat hand primitive are the curvature and closure of each digit, along with the abduction angles. Wrist joints and hand orientation are intentionally excluded, as each primitive is designed to be as flexible as possible. The same flat hand primitive is useful to describe concepts ranging from *stop* to *fry/cook* and *read*. All defined dynamic primitives are listed in Table 6.3, and static primitives in Table 6.4 indicating the source of feature information used to train and recognize each primitive.

A sample set consisting of user-dependent labelled primitives was used for training and evaluating the performance of each HMM using distinct training and testing subsets. A 4-state continuous model with 3 mixtures and a full covariance matrix was used to model each dynamic primitive. By design, there are only a small set of relatively basic primitives defined for each input channel. During normal interaction, it is important to not only identify likely primitives, but also to identify the presence or absence of any primitive. As mentioned in Chapter 3, a noise model is trained for each unique input channel to help detect the presence or absence of a known primitive. When the noise model is more likely to explain a set of observations than any primitive with the same input channel, the presence of a primitive is unlikely. This noise model



(a) Lip - Left Endpoint (x)

(b) Lip - Left Endpoint (y)

(c) Lip - Right Endpoint (x)

(d) Lip - Right Endpoint (y)

Figure 6.9: Q-Q Plots of Human-Generated Noise in Selected Face Keypoint Measurements

Table 6.3: Dynamic Hand Gesture Primitives

	Dynamic Primitive	Source
HD0	Curling fingers	Finger curvature, finger closure
HD1	Extending fingers	Finger curvature, finger closure
HD2	Forming a pinching grasp	Ftip-thumb proximity, curvature, closure
HD3	Releasing a pinching grasp	Ftip-thumb proximity, curvature, closure
HD4	Index finger curling	Index finger curvature, index finger closure
HD5	Thumb extending	Thumb curvature, thumb closure
HD6	Large clockwise palm rotation	Palm roll
HD7	Large counterclockwise palm rotation	Palm roll
HD8	Large vertical upward movement	Shoulder proximity
HD9	Large vertical downward movement	Shoulder proximity
HD10	Small vertical upward movement	Shoulder proximity
HD11	Small vertical downward movement	Shoulder proximity
HD12	Movement out from chest	Shoulder proximity
HD13	Small circles parallel to floor	Shoulder proximity, Palm direction
HD14	Small circles perpendicular to floor	Shoulder proximity, Palm direction
HD15	Nonvertical movement toward face	Shoulder proximity
HD16	Nonvertical movement away from face	Shoulder proximity
HD17	Small horizontal left movement	Shoulder proximity
HD18	Small horizontal right movement	Shoulder proximity

Table 6.4: Static Hand Gesture Primitives

	Static Primitive	Source
HS0	Cupped fingers	Finger curvature, finger closure
HS1	Pinching grasp	Fingertip-thumb proximity, closure, curvature
HS2	Fully curled thumb	Thumb curvature, thumb closure
HS3	Fully curled index finger	Index curvature, index closure
HS4	Fully curled middle finger	Middle curvature, middle closure
HS5	Fully curled ring finger	Ring curvature, ring closure
HS6	Fully curled pinkie finger	Pinkie curvature, pinkie closure
HS7	Partially curled index finger	Index curvature, index closure
HS8	Extended index finger	Index curvature, index closure
HS9	Extended pinkie finger	Pinkie curvature, pinkie closure
HS10	Extended thumb	Thumb curvature, thumb closure
HS11	Palm facing left	Palm orientation
HS12	Palm facing right	Palm orientation
HS13	Palm facing out	Palm orientation
HS14	Palm facing body	Palm orientation
HS15	Index MCP above palm	Palm direction
HS16	Index MCP left of palm	Palm direction
HS17	Palm directly between index MCP and body	Palm direction
HS18	Hand near head	Shoulder proximity
HS19	Hand in front of chest	Shoulder proximity
HS20	Hand above head	Shoulder proximity

was trained using samples of primitives that do not share common input channels with the noise model.

Please refer to Figure 6.10 for training curves using both traditional HMM and FHMM approaches. It is interesting to note the minor differences in training performance between these two approaches, with the FHMM approach converging with slightly fewer iterations than the HMM approach on average. Recognition performance for each dynamic and static primitive is summarized in Table 6.5. In Table 6.5, the % correct column is calculated as the total generated primitive i correctly recognized as i a percentage of the total generated primitives for i. The % false positives is the total primitives $\neq i$ recognized as i as a percentage of the total number of primitive samples. The % false negatives column is calculated as the number of generated i primitives recognized as a different primitive $\neq i$ as a percentage of the total generated primitives for i. It should be noted that the % correct and % false negatives often do not sum to 100%, as primitives recognized as noise (not detected) are not considered false negatives.

Although the accuracy of this set of primitives is lower than desired (especially since they are user-dependent), the flexible nature of this approach allows the model topologies or even the recognition approach to be interchanged with viable alternatives in future work to improve overall performance. The importance of these results is in the fact that a less scalable, low-level recognition process to identify larger patterns is not required. A small set of basic, but concurrent primitives can be valuable in combination with maintainable (human-readable) concept and knowledge base information. One may ask why a forward generative model approach was selected to demonstrate this aspect of the architecture. The primary reason for using a forward generative model is the simplicity of removing, changing or introducing new primitives into the system. A single primitive can be modified without altering or consuming substantial time retraining any of the remaining primitives.

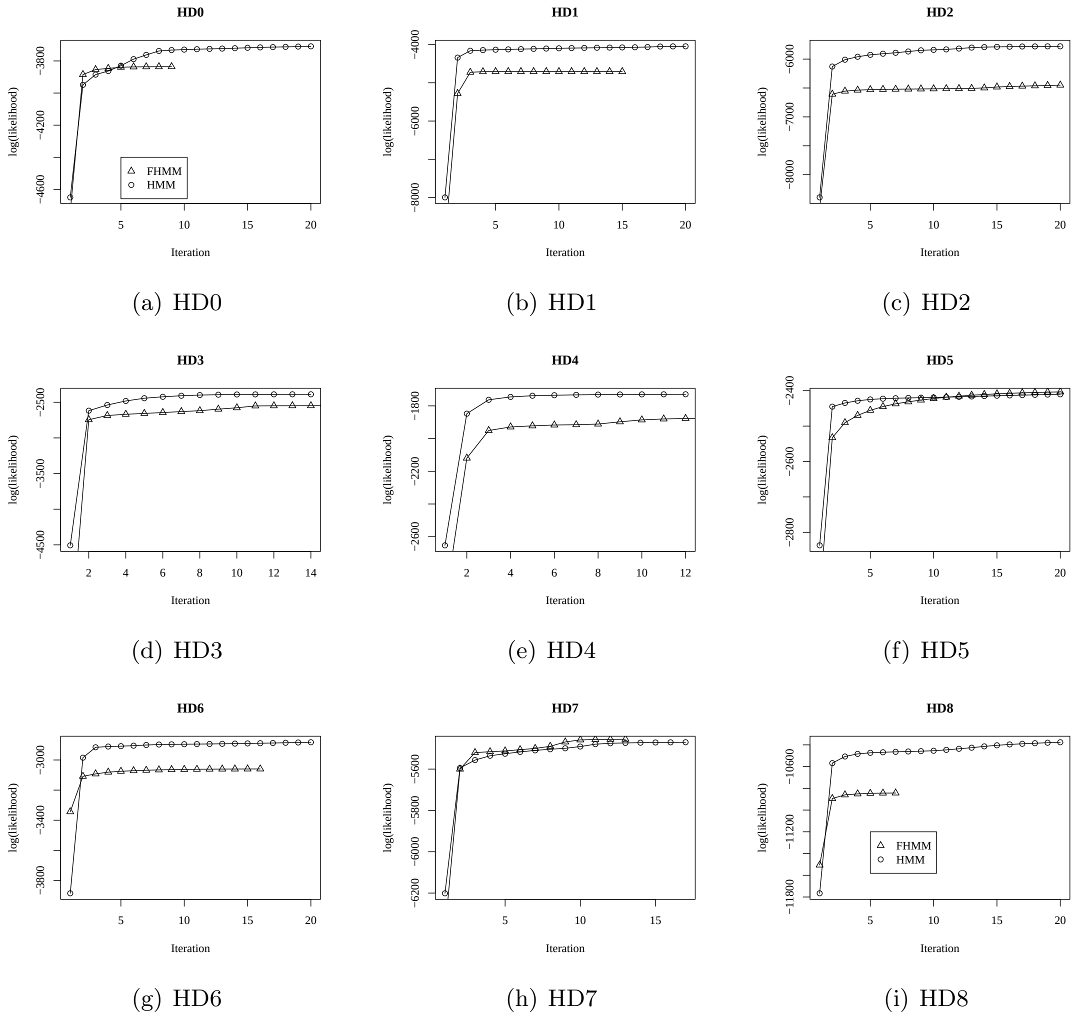
The accuracy of recognizing static primitives is, on average, significantly higher than the dynamic primitives. This outcome does not come as a surprise, as models are build for input data from only one instant in time for the FIS. The dynamic primitives, on the other hand, are more difficult to model and recognize due to their time-varying nature or reliance on sequences of feature vectors with variances in both value and length. It is also interesting to note the relatively low performance of primitives based on shoulder proximity. These results can be attributed to a discrepancy between where the system considers the shoulder, and where the shoulder is physically located in space. This discrepancy can be addressed in future work by sensing the shoulder location and incorporating this information in the perception process rather than an occasional manual calibration of shoulder position. Moving from a sensor glove based data acquisition system to a vision based system will introduce new challenges, but it also provides the means to provide more accurate information about the position of the hand relative to the body (including shoulder proximity).

Variations in the recognition accuracy between different dynamic primitives can also be explained by the application of a single left-right topology to model each dynamic primitive. In this research, the transition, emission, and initial state parameters of each model are adjusted during training. However, the topology itself is also an important parameter that can be used to further optimize accuracy. Incorporating support for different topologies, each designed for a specific dynamic primitive, can result in models that are more representative of their corresponding primitives. A similar approach can be applied to build not one, but multiple noise models with different topologies that may more accurately represent different forms of noise in available channels. It should be noted that a relatively high level of false positives is preferred rather than a high level of false negatives. This preference is desired since extraneous false positives can be discarded if they are not required when forming concepts. A large set of false negatives, on the other hand, will reduce the set of primitives available to form concepts.

The optimization of HMM topologies to model dynamic hand gesture and facial expression primitives is an interesting area of focus for future research, and can build on existing work in other application areas. Some relevant techniques include use of discriminative information to tune topologies for use in document analysis and handwriting recognition [124–126], and use of statistical properties of symbols with respect to the language to select model length in left-right topologies [127].

6.3.2 Facial Expression Primitives

A process similar to the one performed with the hand gesture primitives was followed to train and evaluate facial expression primitives. Simple, concurrent primitives are defined to allow a subset



(a) HD0 (b) HD1 (c) HD2

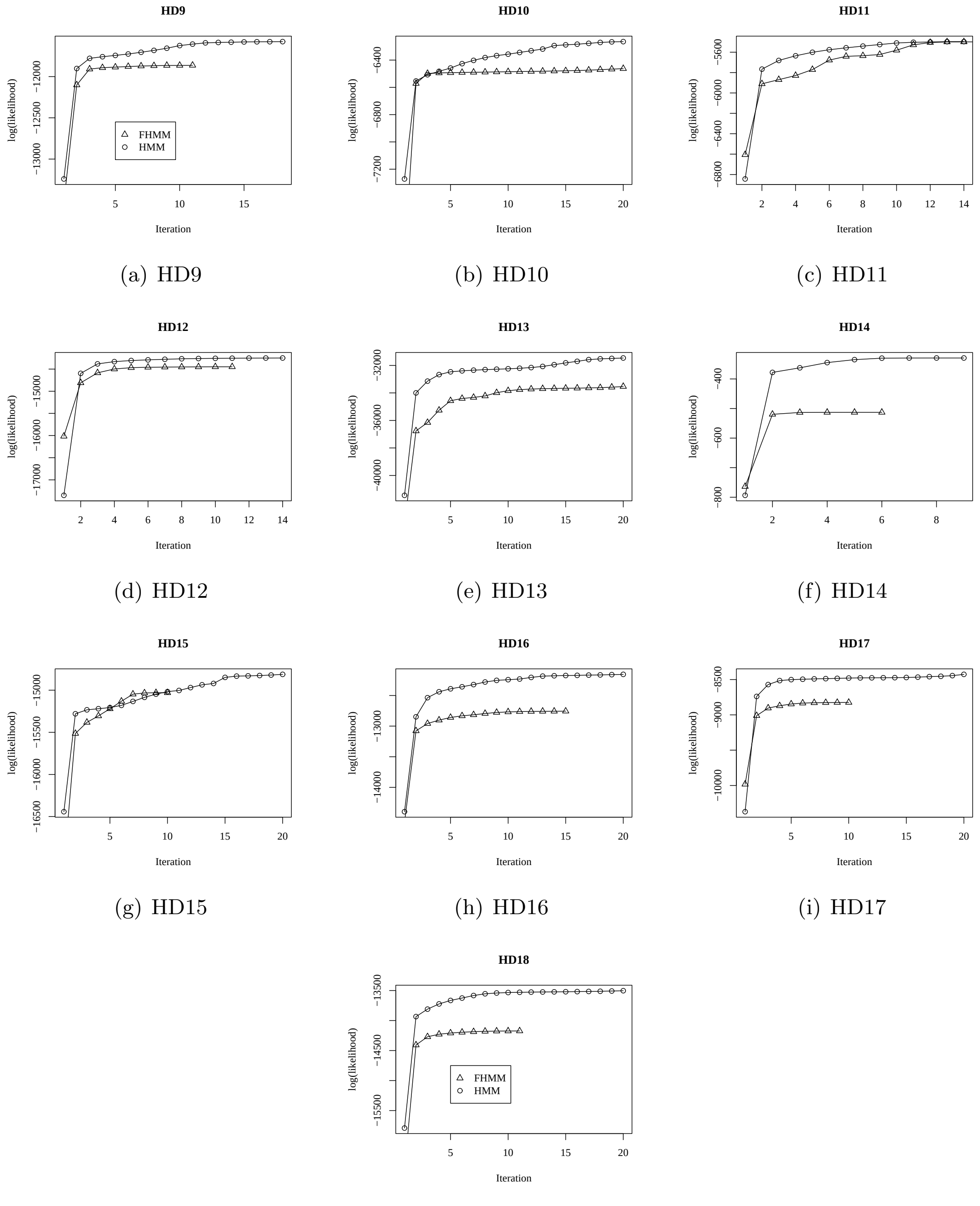
(d) HD3 (e) HD4 (f) HD5

(g) HD6 (h) HD7 (i) HD8

Figure 6.10: Training Characteristics for Dynamic Hand Gesture Primitives (part 1/2)

of the seven basic facial expressions to be described as concepts; happy, sad/mad, shock/surprise and neutral. In addition to these concepts, facial expression primitives are also valuable to describe multimodal concepts including drink, eat, and search. Dynamic primitives and their sources of features are listed in Table 6.6, and static primitives in Table 6.7.

A 4-state continuous model was used to model each dynamic primitive used in these experiments. A sample set of 40 user-dependent manually labelled primitives were used, divided into disjoint testing and training subsets. As shown in Table 6.8, recognition performance is comparable between the traditional and the FHMMs. This similarity is as expected since the same algorithm is used for both hand gesture and facial expression primitive modelling and recognition, varying only in model parameters and feature vectors. Both dynamic and static primitives associated with a small mouth region were less accurate than the other primitives. This result is understandable since a contracted or closed mouth provides only a very small area to sample



(a) HD9 (b) HD10 (c) HD11

(d) HD12 (e) HD13 (f) HD14

(g) HD15 (h) HD16 (i) HD17

(j) HD18

Figure 6.11: Training Characteristics for Dynamic Hand Gesture Primitives (part 2/2)

Table 6.5: Hand Gesture Primitive Recognition Performance

Primitive	% Correct (HMM / FHMM)	% False Positives (HMM / FHMM)	% False Negatives (HMM / FHMM)
HD0	82 / 74	1.6 / 1.5	15 / 18
HD1	89 / 87	1.5 / 0.5	2.5 / 7.7
HD2	76 / 94	0.4 / 0.3	15 / 2.6
HD3	75 / 82	0.6 / 0.8	18 / 5.1
HD4	87 / 89	2.4 / 2.7	7.7 / 7.9
HD5	79 / 82	1.2 / 0.9	10 / 7.9
HD6	62 / 67	0.6 / 0.8	26 / 18
HD7	67 / 76	0.5 / 1.2	18 / 18
HD8	69 / 64	0.4 / 0.7	21 / 25
HD9	64 / 64	0.9 / 0.8	28 / 23
HD10	76 / 64	1.1 / 0.1	23 / 25
HD11	74 / 82	3.8 / 2.9	23 / 13
HD12	76 / 72	0.7 / 0.8	10 / 15
HD13	69 / 59	0.5 / 0.5	23 / 26
HD14	61 / 72	0.8 / 0.7	30 / 18
HD15	69 / 67	0.9 / 0.4	21 / 23
HD16	61 / 64	0.5 / 0.9	25 / 28
HD17	67 / 63	0.2 / 0.4	28 / 33
HD18	69 / 64	0.4 / 0.5	23 / 20
HS0	82	0.5	13
HS1	95	0.6	5.1
HS2	74	0.2	7.7
HS3	82	0.1	7.7
HS4	79	0.5	7.7
HS5	82	0.9	10
HS6	87	0.0	5.1
HS7	85	0.6	10
HS8	87	0.4	5.1
HS9	79	1.0	15
HS10	79	0.4	13
HS11	87	0.6	7.7
HS12	92	0.2	7.7
HS13	82	0.4	13
HS14	90	0.5	7.7
HS15	90	0.5	7.7
HS16	92	0.6	7.7
HS17	79	0.4	13
HS18	77	0.5	15
HS19	72	0.4	10
HS20	74	0.6	15
Mean (Dynamic HMM/FHMM)	72 / 73	0.1 / 0.9	19 / 18
Mean (Static)	83	0.5	9.7

Table 6.6: Dynamic Facial Expression Primitives

	Dynamic Primitive	Source
FD0	Mouth opening	Lips
FD1	Mouth closing	Lips
FD2	Lips extending	Lips
FD3	Lips contracting	Lips
FD4	Eyebrows raising	Eyebrow-Lip Proximity
FD5	Eyebrows lowering	Eyebrow-Lip Proximity

Table 6.7: Static Facial Expression Primitives

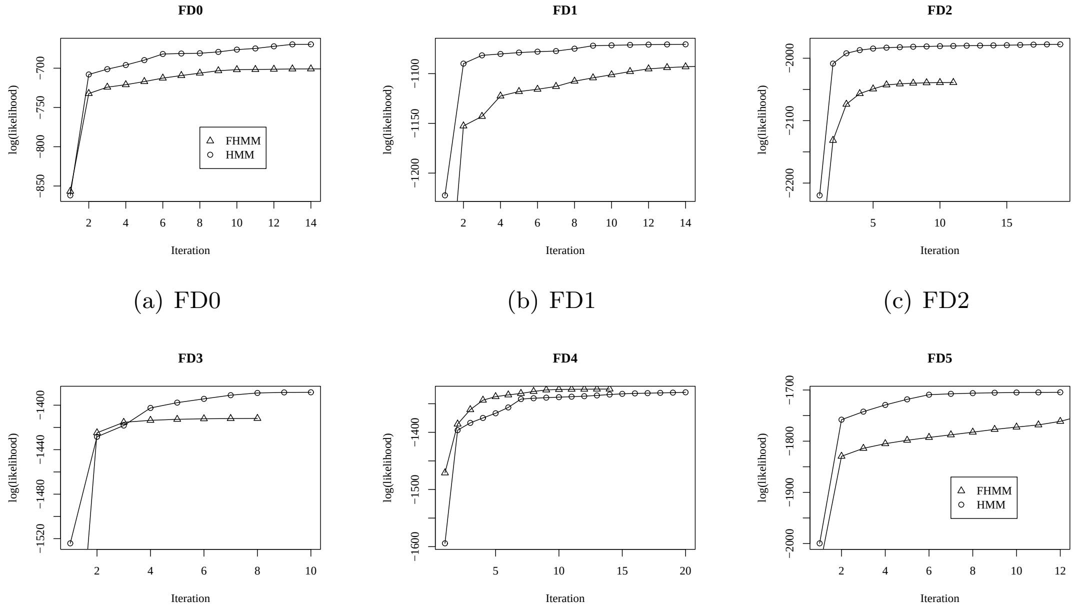
	Static Primitive	Source
FS0	Eyebrows low	Eyebrow-Lip Proximity
FS1	Eyebrows relaxed	Eyebrow-Lip Proximity
FS2	Eyebrows high	Eyebrow-Lip Proximity
FS3	Lips extended	Lips
FS4	Lips contracted	Lips
FS5	Mouth open	Lips
FS6	Mouth closed	Lips

texture and colour information for keypoints. The relatively low performance of FS1 - relaxed eyebrows cannot be attributed to a small sampling area, but can be explained by the difficulty in distinguishing between relaxed and either high or low eyebrow states.

The strength of the FHMM approach lies in its ability to converge in slightly fewer iterations, as previously observed with hand gesture primitives (Figure 6.10). The training performance for both the traditional HMM and the FHMM approach is shown for each dynamic primitive in Figure 6.12. It is interesting to note the difference in rate of convergence between the FHMM and HMM approaches when used to model dynamic facial expression and hand gesture primitives. The FHMM approach typically converges in fewer iterations than the traditional HMM approach. The reduced number of training iterations for the FHMM approach can be explained by its support for more complex interactions between the states and observations than those permitted in a traditional HMM. On the other hand, the quicker convergence can also be due to getting stuck in a local optimum. Since the performance results are comparable (Table 6.8), either approach provides a suitable model of each defined human expression primitive, providing slightly different outcomes for different primitives.

6.4 Ranking Concepts

Many of the defined primitives are insufficient to describe a meaningful concept on their own. The recognition of primitives and formation of concepts were divided in this manner by design, to both maintain a low-complexity search space to recognize primitives, and to easily incorporate and maintain expert knowledge describing each concept in terms of concurrent primitives. When given a set of candidate primitives, it is desirable to select reasonable concepts that explain the



(a) FD0 (b) FD1 (c) FD2

(d) FD3 (e) FD4 (f) FD5

Figure 6.12: Training Curves for Dynamic Facial Expression Primitives

Table 6.8: Facial Expression Primitive Recognition Performance

Primitive	% Correct (HMM / FHMM)	% False Positives (HMM / FHMM)	% False Negatives (HMM / FHMM)
FD0	88 / 76	2 / 3	0 / 12
FD1	76 / 76	3 / 2	12 / 12
FD2	76 / 71	4 / 3	6 / 6
FD3	53 / 65	0 / 1	35 / 12
FD4	88 / 82	1 / 0	0 / 12
FD5	82 / 76	1 / 1	12 / 6
FS0	84	2	10
FS1	68	0	23
FS2	87	1	3
FS3	84	1	10
FS4	77	1	13
FS5	94	2	3
FS6	81	3	10
Mean (Dynamic)	77 / 75	1.8 / 1.6	11 / 10
Mean (Static)	82	1.4	10

presence of as many of the primitives as possible.

Concepts were first evaluated without feedback, relying only on the accuracy of the recognized primitives and quality of the fuzzy descriptions of temporal relationships between primitives. The performance of this knowledge-based integration approach was evaluated using concepts from a manually generated domain specific knowledge base. Sets of aggregate facial expression and hand gesture primitives were generated using visual prompts providing the definitions of each concept, or the required set of primitives. Each concept is briefly described in and related to the previously defined primitives in the following list. It should be noted that these concept descriptions are intentionally in human readable form to emphasize the relevance of existing annotation systems (i.e. HamNoSys) to describe concepts in a form that can directly make use of recognized concurrent primitives.

glass / cup / drink container The hand forms a shape as though it holds an imaginary glass: cupped hand in front of chest, oriented with palm facing left and thumb at top (all static primitives).

pick up / select / start move Fingers grasp an imaginary item from above, and move up slightly while holding the same grasp: pinching (a dynamic primitive) with hand in front of chest, oriented with palm facing down, followed by holding a pinch configuration during a rising hand motion.

table Flat hand, palm facing down, followed by curling of the digits to symbolize legs of the table.

locate / search Cupped hand in front of face (eye), oriented with palm facing left and thumb closest to face, eyebrows extended (raised)

speed / power / energy Curled index finger touching thumb followed by thumb extension and further curling of index finger.

high / large / fast Flat hand, palm facing down, located above head.

growth Flat hand, palm facing down, moving up from chest.

low / small / slow Flat hand, palm facing down, moving down from chest.

go Extended index finger, moving in circular motion away from chest

room Flat hand, palm facing left and thumb at top while hand moves down - followed by flat hand, palm facing body, thumb at top while hand moves down

prepare / make Fist in front of chest with small down and up movements.

clean Flat hand, palm facing down while hand makes circular motions parallel to floor.

drop off Reverse of pick up: Fingers in grasping configuration move slightly down while releasing the grasp.

morning / sunrise Flat hand, palm facing up while hand makes arc motion toward face

evening / sunset Flat hand, palm facing up while hand makes arc motion away from face

spaghetti Fist shape except pinkie finger extended while hand makes small circular movements and hand moves out from chest

bread Flat hand, palm facing left, thumb toward body while hand makes small down and up movements (cutting slices)

cook / toast / fry Flat hand, palm facing up followed by rotation along the forearm axis so the palm faces down, followed by a reverse rotation until the palm faces up

cat / whiskers Fingers in pinching configuration, palm facing left, hand moves left and right at side of face while lips are extended.

eat / feed Pinched fingers move in arc toward face (mouth) while mouth opens

tomorrow / future Fingers curled with thumb extended move straight out from face

today / now Fingers curled with thumb extended move straight down from face

yesterday / past Fingers curled with thumb extended move straight back from face (move behind face)

drink Cup shape formed by fingers before hand follows arc toward face (mouth) while mouth opens

close Flat palm, facing to the left with fingers pointing up rotates to face away from body

open Flat palm, facing away from body with fingers pointing up rotates to face left

window Flat palm, facing toward body, moves slightly up and down

happy Lips extending, eyebrows relaxed

sad Lips contracted, mouth closed, eyebrows low

shock Mouth open, eyebrows high

neutral Lips relaxed, eyebrows relaxed

stop Flat palm, facing away from body, eyebrows lowering

The accuracy of forming a concept using simple, concurrent primitives and human maintainable concept definitions was evaluated by soliciting expressions from the user with random visual prompts for concepts. Previously trained models for each primitive (and the noise models) were used to provide the best primitives for each channel ending at each time step. This information was used, as described in Algorithm 4.1 to arrive at a ranked list of concepts. The accuracy of this process using the provided set of concepts and primitives is summarized in Table 6.9 using 8 samples of each isolated concept. The two % correct columns in this table are significant, as the first column considers a concept to be correct only if it is ranked #1, whereas the second column will consider a concept to be correctly matched if it is found in any of the top 3 positions. Obviously, the % correct figure should increase as the number of positions considered is increased, however, only a finite set (3 in this scenario) of concepts are passed along to the next stage in the process to maintain a reasonable search space for hypotheses of intent. At first glance, the average accuracy may appear lower than expected, especially since these are isolated concepts, and additional constraints have been introduced when forming concepts from primitives. Although additional constraints are introduced, multiple primitive candidates at each time interval are being considered and compared against concept definitions, whereas the performance of just a single primitive was being evaluated in Table 6.5 and Table 6.8.

The impact of leveraging conflicts can be observed in the final two columns in Table 6.9, where feedback from mode conflicts are included. These results were generated using $w_{unexplained} = 0.5$ initially, with a linear decreases to 0.25 over the average concept duration. Similarly, $w_{relevance} = 0.25$ initially, and increases linearly to 0.5. $w_{overlap}$ also changes linearly from 0.25 to 0.5 to emphasize exploration or divergence at the beginning of the process and emphasize convergence (minimize overlap) as time progresses. As can be observed in Table 6.9, on average, the accuracy of identifying individual concepts is improved when leveraging feedback from mode conflicts using $\alpha_p = 0.8$. It is likely that $\alpha_p = 0.4$ is adversely impacting performance by relying too heavily on concept definitions, discounting the valuable information provided in observed primitives.

To explore the robustness of concept formation against noise, normally distributed signal noise was used to control the quality of input primitives, and observe the effect of noise on the performance of integration. As shown in Figure 6.13, and Table 6.9, considering the top N concepts increases the probability that the correct concept will be considered when attempting to understand the underlying intent. Even though the accuracy decreases rapidly with an increase in signal noise, the human knowledge used to distinguish one concept from another helps to compensate for the (artificially) reduced input quality.

6.5 Forming Hypotheses of Intent

Since the purpose of the presented system is to assist devices toward understanding the intent of human expression, the accuracy of the hypotheses of intent formed by a device is a critical measure of performance for the end user. During interaction, the human participant does not directly observe the performance of the concurrent primitives, nor do they observe the process of identifying sets of concepts. Only the resulting intent is used to perform an action or provide the user with some other form of feedback.

The sample set of scenarios used in experiments to illustrate the closed-loop architecture is described in this section. To aid human readability, the linear form of conceptual graphs is used (CGLF) for each of the described scenarios. In practice, this knowledge must be adapted to suit the domain in which robot-interaction will occur, and to provide sufficient knowledge

Table 6.9: Accuracy of Concept Formation

Concept	% Correct (top 1)	% Correct (top 3)	% After Mode Conflict $\alpha_p = 0.8$	% After Mode Conflict $\alpha_p = 0.4$
glass	75	88	75	75
pickup	100	100	100	100
table	75	100	75	88
locate	75	100	75	100
speed	75	100	75	100
high	62	75	62	75
growth	88	100	88	100
low	88	100	88	100
go	62	62	62	62
room	75	88	75	88
make	62	75	62	62
clean	100	100	100	100
dropoff	100	100	88	100
morning	50	75	50	62
evening	75	88	75	88
spaghetti	62	75	62	62
bread	62	75	50	62
cook	88	88	88	88
cat	75	75	75	75
eat	75	75	75	75
future	38	62	38	62
present	88	100	88	88
past	100	100	100	100
drink	75	88	62	88
close	88	100	88	88
open	75	88	75	88
window	62	75	50	62
happy	75	75	75	75
sad	88	100	88	88
shock	62	75	50	62
neutral	88	100	75	100
stop	62	75	62	62
Mean	76	88	73	82

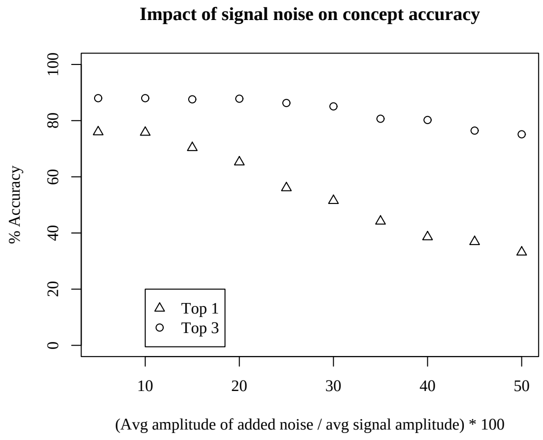


Figure 6.13: Average concept accuracy in the presence of noisy primitives.

that encompasses the robot's available functionality. Although building an appropriate base of knowledge can be time consuming, the process is required in one form or another when developing the specifications for a robot's actions. The same information can be re-used to not only associate a task or action with a known scenario, but also to tie together human perception with known scenarios to help infer intent. A base of existing relevant relationships in the form of a concept and relation hierarchy is more general, requiring less customization for different application domains.

```
A0: [glass:a]<-(object)<-[pick up]->(agent)->[robot:mag1]
    [table]<-(loc)<-[glass:a]

A1: [go] -
       (agent)->[robot]
       (destination)->[kitchen]
       (attr)->[fast]

A2: [kitchen]<-(loc)<-[table]<-(object)<-[prepare] -> (agent)->[robot:mag3]

A3: [clean] -
       (agent)->[robot:mag3]
       (object)->[table]->(location)->[kitchen]

A4: [deliver] -
       (object)->[note]->(attr)->[small]
       (agent)->[person:Fred]
       (destination)->[robot:mag1]
```

```
A5: [prepare] -
       (agent)->[robot]
       (object)->[breakfast]
       (time)->[morning]

A6: [prepare] -
       (agent)->[robot]->chrc->[location]->[value:athome]
       (object)->[bread]
       (instrument)->[toaster]->(location)->[kitchen]
       (time)->[morning]

A7: [prepare] -
       (agent)->[robot]->chrc->[location]->[value:athome]
       (object)->[spaghetti]
       (instrument)->[saucepan]->(location)->[cupboard]
       (time)->[evening]

A8: [feed] -
       (object)->[cat]->(owner)->[person:Fred]
       (agent)->[robot]
       (time)->[today]

A9: [deliver] -
       (object)->[drink]-
               (attr)->[cold]
               (owner)->[person:Sally]
       (agent)->[robot]
       (destination)->[fridge]
       (time)->[evening]

A10: [go] -
       (time)->[morning]
       (agent)->[robot:mag3]
       (destination)->[kitchen]

A11: [robot]<-(agent)<-[close]->(object)->[door]->(position)->[north]

A12: [robot]<-(agent)<-[close]->(object)->[window]->(position)->[north]

A13: [deliver] -
```

```
        (object)->[drink]
        (agent)->[robot]
        (destination)->[person]->(attr)->[thirsty]

A14: [deliver] -
        (object)->[drink]
        (agent)->[robot]
        (destination)->[table]->(loc)->[kitchen]

A15: [stop]->(agent)->[robot]
```

Since the resulting intent is important to any human participant, its accuracy is a critical indicator of the presented architecture. Hypotheses of intent were evaluated in a similar manner to individual concepts. Complete phrases were solicited from the user using visual prompts, randomly based on any of the previously defined concepts. The previously trained primitives were used to identify primitive candidates that were subsequently used to identify concepts, and finally form a hypothesis of intent using the sample knowledge base.

Unfortunately, determining if the identified hypothesis is correct (or close enough) can be subjective. To ensure performance is evaluated as objectively as possible, a selected hypothesis of intent is considered correct if and only if no other analogy is closer to the manually requested intent. The measure of similarity to determine accuracy is given in the approach discussed in Chapter 4. Results are summarized in Table 6.10, using 9 samples of each intent. Average accuracy should improve over individual concepts since domain specific knowledge is incorporated to help constrain possible outcomes. Unfortunately, the similarity between cases in the knowledge base sharing common concepts has the reverse effect by introducing additional ambiguity, and an increase in complexity moving from isolated concepts to full sets of concepts. The effect of feedback from temporal conflict detection and resolution is particularly evident in Table 6.10, where the average accuracy of the selected hypothesis of intent noticeably improves. The effect of α_c is similar to α_p, suggesting that smaller values of α_c and α_p should be used to ensure new information in primitives and concepts is not inadvertently overlooked.

6.6 Implementation Issues

Several interesting issues were encountered during the implementation of the multimodal architecture. These issues included transient network connectivity between (mobile) devices, in addition to control and data acquisition related obstacles. This section describes some of the interesting issues and how they were approached during the implementation.

Table 6.10: The accuracy of formed hypotheses of intent both relying solely on concepts, and when leveraging temporal conflicts.

Intent	% Correct	% After Leveraging Conflict	% After Leveraging Conflict
		$\alpha_c = 0.8$	$\alpha_c = 0.4$
A0	89	78	100
A1	89	56	89
A2	78	67	78
A3	100	67	100
A4	78	56	100
A5	100	78	100
A6	78	67	78
A7	78	89	89
A8	89	89	89
A9	67	67	78
A10	67	67	67
A11	33	22	44
A12	67	67	78
A13	100	67	100
A14	78	67	89
A15	78	67	89
Mean	79	67	86

6.6.1 Timing

From a practical perspective, the ability of a robot or other device to understand human expression is most valuable when it can be accomplished in a reasonably short time period. A robot that takes an excessive period of time to respond or to carry out a task can result in unnecessary waiting and an inefficient use of human time. One can determine how the system scales with a larger set of primitives, concepts, and knowledge base by analyzing each of the algorithms used in this system. This information is brought forward from previous chapters and summarized here for completeness. Starting from perception, the feature histogram based mean-shift approach scales linearly with the number of pixels being considered and the number of scale variations for each feature. With feature information available, the recognition process for dynamic primitives scales proportional to the square of the number of states in each primitive, and linearly with the number of observations and number of primitives. Forming concepts requires time proportional to the cube of the number of time steps being considered, and scales linearly with the number of defined concepts. Since an approximate reasoning approach is used to form hypotheses of intent, the required time is not necessarily related to the number of concepts, and can be adjusted to balance available time against desired accuracy. It is important to note that each component of this system currently scales linearly with the relevant inputs of features, primitives, or concepts.

Scalability may be helpful when expanding the set of concepts and primitives, but it is also valuable to provide empirical measurements for a specific implementation. Using the primitives, concepts, and hypotheses described in this chapter, on a Pentium 4 2.26Ghz (4490 bogomips), approximately 100ms is required to identify features in each image, 15ms to identify each set of primitive candidates, and 20ms to form each set of concepts. Up to 100ms is allocated to form hypotheses of intent using the approximate reasoning approach and to provide conflict feedback. As previously discussed, the maximum tolerable latency for human interaction is reported as either a maximum latency of 50ms [113], or maximum standard deviation of no more than 82ms [114]. Although the time required for this specific implementation and hardware platform is longer than desired for interactivity, it is important to note that alternate implementations for individual aspects of the system can be integrated into the same architecture in the future to address application specific interactivity constraints.

6.6.2 Inter-Process Communication Framework

Robust communication is essential to share information about human expressions between all components in a practical human-robot interaction system. Data acquisition, integration, understanding, and the physical actions may be handled by physically distinct devices, but must still share information in a timely and robust manner. Several common distributed software frameworks are robust in theory, but are often implemented in a manner not suitable for robust communication. Unfortunately, most software-based remoting implementations and high-level distributed frameworks, including popular Common Object Request Broker Architecture (CORBA) based implementations, assume the underlying network connectivity is reliable and persistent. When the devices in a distributed system are traditional personal computers on a wired or relatively stationary wireless connection, the assumption of persistent connectivity is reasonable. On the other hand, when dealing with mobile devices such as a domestic service robot communicating over widespread general purpose wireless networks (i.e. 802.11a/b/g), connectivity cannot be guaranteed. Not only is connectivity not guaranteed, but network identity (i.e. IP address) is also not guaranteed to remain consistent over time in some environments.

Several mobility issues are being resolved with the support of emerging addressing and wireless communication protocols. However, until these protocols are in widespread use, there is a need to transfer simple messages between the distributed components of a human robot interaction system. Most of these simple messages are relevant only over a short period of time, and do not require the resource overhead or complexity of persistent storage. To maintain flexibility, and to support the need to introduce new observation sources for human expression and devices to carry out actions, the topology and even the sources and destinations of messages should not be tied to a specific device or application.

The author implemented a lightweight messaging approach using a subscriber-publisher pattern to separate communication from both physical network connectivity and from specific devices and applications. All messages between each component in the implemented system are transferred over virtual communication channels similar to the one shown in Figure 6.14. This abstraction cleanly isolates connectivity, recovery, negotiation, and queueing from the application logic. Human-robot interaction modules handle incoming messages and send out results without

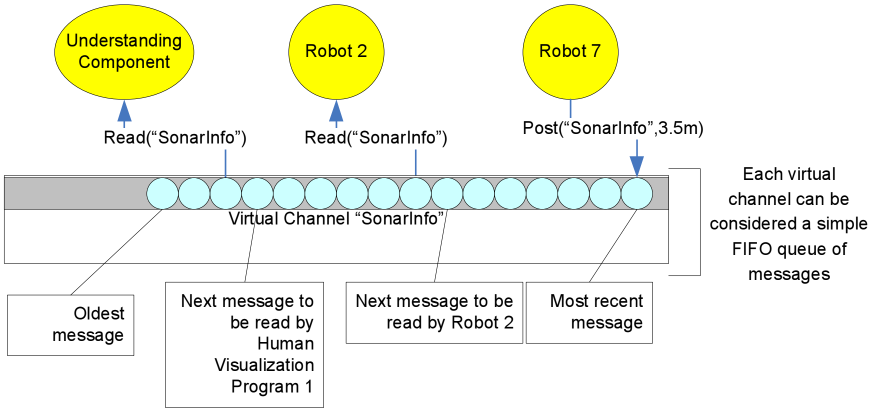

Understanding Component
Robot 2
Robot 7
Read("SonarInfo")
Read("SonarInfo")
Post("SonarInfo",3.5m)
Virtual Channel "SonarInfo"
Each virtual channel can be considered a simple FIFO queue of messages
Oldest message
Next message to be read by Human Visualization Program 1
Next message to be read by Robot 2
Most recent message

Figure 6.14: Channel-based messaging supports robust communication over unreliable links between each loosely-coupled component in the implemented system

concern about error recovery, connectivity, or even the number or location of recipients.

In practice, these virtual channels can be considered equivalent to, and handled in the same manner as, a simple first-in-first-out queue from any device. Messages can be added from one or more components on any device, and are available to all subscribers. Temporary loss of connectivity and changes in network configuration are handled using auto-discovery and reliable message recovery. For example, when a mobile robot roams between wireless access points not designed for mobile clients (i.e. 802.11a/b/g), connectivity may be lost and its IP may change. The robot continues to transparently add messages to the virtual channel during this transition. As soon as connectivity is re-established, all valid pending messages become available to other participants using the virtual channel.

The advantage of this loosely-coupled messaging framework over a framework with a directory service or a central repository is in its ease of deployment and maintenance. Since no knowledge is required about the physical network, components in the multimodal architecture can be seamlessly transferred from co-existing on the same machine to existing on multiple distinct machines without the need to reconfigure modules. No central directory service eliminates the single point of failure should connectivity be lost with a directory service. The lack of physical device configuration information is helpful for supporting the dynamic addresses assigned to the mobile robots during roaming in addition to providing support for transparent monitoring and diagnostics.

6.6.2.1 Topology Changes

Each node in the system maintains an active list of directly reachable nodes. This list is dynamically updated based on changes in network topology. When a new node is introduced, it blindly notifies all of its neighbours about its presence. When a notification is received by any node, it is used to establish a direct, persistent connection to exchange messages. This notification process is repeated at periodic intervals in case the initial startup notification was unsuccessful for any reason. Failure to read or write information across a previously established connection results in its immediate disconnection. This may occur when a mobile robot is transferred from one access point to another, and cannot be avoided when a new IP is assigned to a device.

The drawback to an auto-discovery process rather than a dedicated naming service is the need to provide unique identifiers for each component in the system. These unique identifiers cannot be associated with a network address, nor with a device without limiting the mobility of components. It was decided that manual provisioning of unique identifiers for each component was an acceptable trade-off to eliminate the need to deploy and maintain a naming service for this implementation.

Since the typical use of distributed components in this research involves many more components than devices, the implementation minimizes external direct connections to at most one per pair of devices rather than one per pair of components. When multiple nodes exist on the same device, they transparently share a single external port. Nodes on the same device negotiate for control of the port (a limited resource) using a simple but robust first-to-acquire approach. The node with control of the external port handles external notification for the device and routes messages between nodes on remote devices and nodes on the local device. If communication with this routing node fails for any reason (i.e. module shut down, software failure), all local nodes renegotiate for control of the external port and resume communication.

6.6.2.2 Recovery

Recovery is a critical aspect of a robust inter-process communication framework. Failures are inevitable, whether in the physical network or in software, and must be gracefully handled to minimize the impact on human-device interaction. When a connection is established between two nodes, they first exchange subscription information. As soon as a local node is aware of the channels to which a remote note is subscribed, the local node reviews its buffered old messages and provides the remote node with any messages it may have missed.

Each message is tagged with a real time-to-live value to indicate is useful lifespan. For example, a message providing hand configuration information during human-device interaction is useful for no more than about 500 ms. This time-to-live value is used to discard old messages as soon as they are no longer needed. Only messages that are still active may be used for recovery. This message expiry approach is used to keep resource consumption manageable during network disruptions without the additional overhead or complexity of persistent storage.

To prevent an overwhelming number of messages from being sent out after momentary failures, an attempt is made to include only messages that have not been previously received or processed are considered for recovery. When a message is received or processed, an acknowledgement is returned to the originator of the message. Since multiple recipients may exist for any given channel, multiple acknowledgements are maintained and associated with each unique recipient. Unfortunately, identification of a recipient presents a problem when the device address and network topology is dynamic. In this implementation, a manually generated unique name is used for each node to assist during recovery. This is not an ideal solution, as it relies on the implementer to ensure node names are unique and provides little recourse if two nodes share the

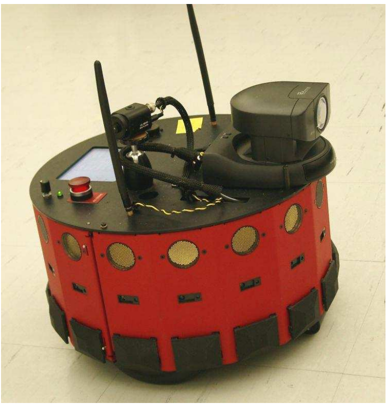


Figure 6.15: The MagellanPro mobile robot hardware platform provides actuators and sensors to carry out specific actions based on an understanding of human expression.

same name. Unfortunately, some unique identifier is required for each node to efficiently recover messages.

Messages may be buffered at intermediate nodes, either locally or anywhere across the network. These buffering nodes provide sources for message recovery in case the originating node fails, losing its own buffer of old, but active messages. Since the network topology is not specified in this inter-process communication scheme, and messages can be buffered by any node, it is likely that recovery will occur from more than one source simultaneously. Unfortunately, this also means that duplicate messages may be received during recovery. Duplicate messages are detected and discarded in a similar manner to transmission control protocol (TCP): a unique sequence number is generated and included as part of the header when a message is originally created. Recent ranges of sequence numbers, together with originating source, are maintained by the recipient. Duplicate messages are detected by quickly looking up the sequence number and originator of each message.

6.6.3 Low-Level Robot Control and Perception Framework

The MagellanPro[1] mobile robot hardware platform is used to carry out actions and provide responses during natural interaction. This hardware platform, as illustrated in Figure 6.15 consists of a x86 based processor to run low and high-level control logic and is connected to several FPGAs that control sensors and actuators. It is a general purpose research robot platform that is well suited to carry out navigation and simple environment modification tasks representative of simple domestic service robots.

Preliminary experiments during the early stages of this research encountered several obstacles related to low-level robot control and perception. These issues included the inability to access raw sensor data and characteristics, in addition to unreliable latencies between sensing and availability of sensor data in an application. These latencies were most noticeable in sonar data,

[1] MagellanPro is a registered trademark of Real World Interfaces, Inc.

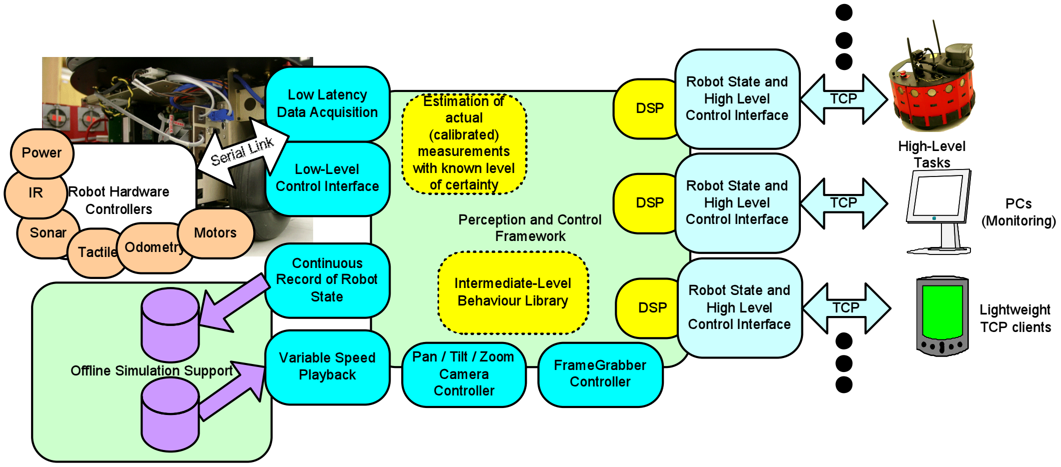

Power
IR
Robot Hardware Controllers
Sonar
Tactile
Odometry
Motors
Serial Link
Low Latency Data Acquisition
Low-Level Control Interface
Continuous Record of Robot State
Variable Speed Playback
Offline Simulation Support
Estimation of actual (calibrated) measurements with known level of certainty
Perception and Control Framework
Intermediate-Level Behaviour Library
Pan / Tilt / Zoom Camera Controller
FrameGrabber Controller
DSP
DSP
DSP
Robot State and High Level Control Interface
Robot State and High Level Control Interface
Robot State and High Level Control Interface
TCP
TCP
TCP
High-Level Tasks
PCs (Monitoring)
Lightweight TCP clients

Figure 6.16: A flexible, low-latency multi-robot control and perception framework provides the access required to the physical robot hardware to carry out physical actions and provide valuable cues about the robot's environment.

where the time at which a reading was received was accurate to within approximately 240ms, an unacceptable level of uncertainty.

The author designed and implemented a new control and perception framework to address both the latency and low-level access issues. This framework was designed to provide a lightweight and transparently distributed data acquisition and control layer for either single or multiple robots. The utility of this framework has been observed in applications outside the scope of this thesis, including providing the action layer for a multi-agent architecture for mobile robots [128].

This flexible software platform provides full access to all sensor and actuator functions in addition to supporting a low-level library of actions and connectivity to local or remote components, such as the high-level human expression understanding module, or a future planning module. The design focuses on making the robot platform functionality available to control applications with minimum implementation effort. Methods are provided to support polled or event-driven control strategies from one or more local or remote applications developed in C++, Java, or a telnet session for quick tests.

With access to raw sensor values, it is possible to analyze this information to extend the usable range of each sensor beyond the region that might be a good fit to a simple function. As shown in Figure 6.18, the raw sonar values are linear across their usable region, with negligible variations between sensors. The only calibration information used is an offset and slope determined by linear regression. The infrared sensors, on the other hand, produce a highly non-linear response as shown in Figure 6.17. In addition to the non-linearity, substantial variations exist between each sensor, especially at the extreme ranges of each sensor's capabilities. Rather than finding a parametric fit for most of the usable region, a simple lookup table was built for each of the 256 raw values on each of the 16 infra-red (IR) sensors using linear interpolation where required. A non-parametric approach ensures the full range of sensor values provide useful calibrated values,

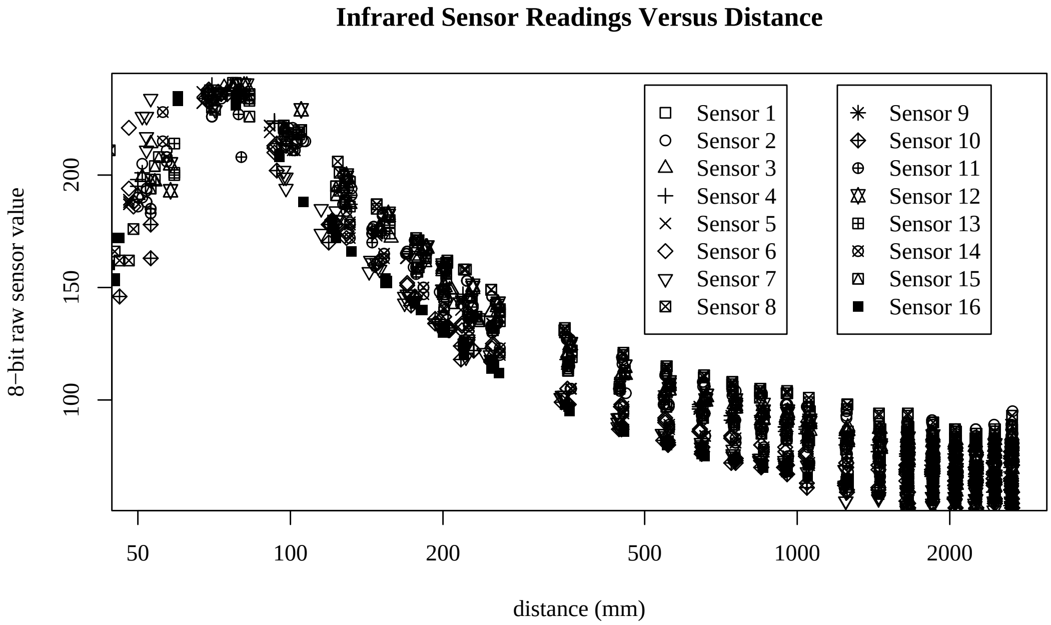


Figure 6.17: Infrared sensors were recorded against odometer values, and calibrated using a non-parametric representation for each of the 16 sensors on each robot.

even where the sensor value is outside a region that can be easily described in a simple function. The calibration curves were obtained automatically by placing the robot perpendicular to a flat surface, and activating the self-calibrate action. This action advances the robot forward until the perpendicular surface is reached (detected using a tactile sensor), travels to preset distances from the surface, and records sensor readings.

This lightweight mobile robot framework can be used in combination with the implementation of the multimodal architecture discussed at the beginning of this chapter to provide a complete interactive system. Once a hypothesis of human intent is found, this hypothesis can be used in future work to identify the desired goal of a robot, and break this goal into simple low-level actions. Some of the actions currently implemented and available as part of the control framework include include *goto, rotate, self-calibrate* and *find-barcode.*

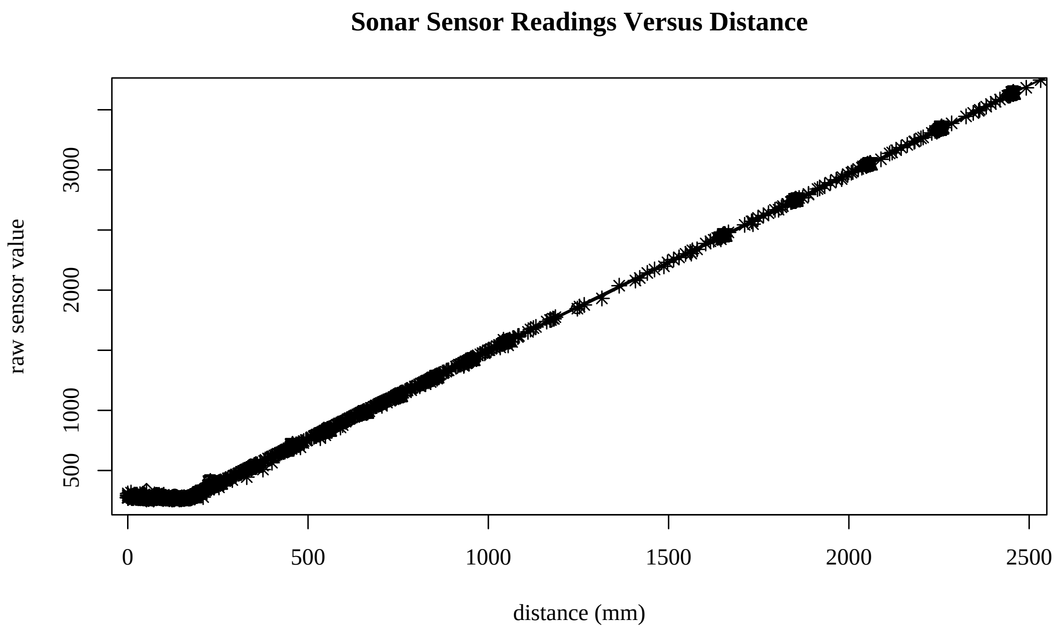


Figure 6.18: Sonar sensors were recorded against odometer values, and calibrated using a linear regression for each of the 16 sensors on each robot.

Chapter Seven

Conclusions and Future Directions

Imagine an environment in which people can easily express their intentions to nearby intelligent devices without the need to prepare and use potentially cumbersome tactile interfaces. Imagine an environment in which humans can focus on their strengths, investing their valuable time and energy in creative activities rather than spending this effort translating their intentions or repeating their communications using tedious interfaces. The need is rapidly growing for an environment or means to efficiently express human intent to nearby devices. It is only a matter of time before intelligent devices and domestic service robots pervade every aspect of human life, becoming essential tools for both work and play for everyone from newborns to the elderly.

Even though a symbiotic relationship already exists, binding people together with their favourite mobile and fixed devices, an intuitive method is still required to reduce the time and effort invested to communicate human intent to these current devices and to future domestic service robots. The research presented in this thesis focused on bringing devices one step closer to understanding human expression. By enabling devices to perceive and take advantage of neglected forms of natural human expression, humans will be able to, one day, interact with domestic service robots and other devices as easily as, or even easier than, interaction with one another.

The presented approach for understanding human expression for human-robot interaction focused on a unified architecture for arriving at a hypothesis of intent based on multimodal human expression. This architecture brought together a perception subsystem (Chapter 3), an integration and understanding subsystem (Chapter 4), and a conflict resolution subsystem (Chapter 5) in a flexible overall framework. An implementation of this framework was presented using facial expressions and hand gestures as a sample subset of the diverse but often overlooked forms of human expression. Modality specific algorithms have been intentionally decoupled from the overall system to emphasize the need to consider higher level concepts and the domain specific knowledge required to explain the underlying intent in any form of multimodal expression. This loose-coupling of components provides the necessary flexibility to replace individual algorithms, to add and remove modality specific processing, or to enhance the system with device specific planning and action components. A flexible approach ensures future extensions can be easily incorporated, encouraging continued use of this framework in future research.

The contributions of this research corresponding to the primary objectives identified in Section 1.2 are summarized in Figure 7.1. The closed-loop architecture for flexible multimodal natural human-robot communication leverages the simplicity and scalability of defining and rec-

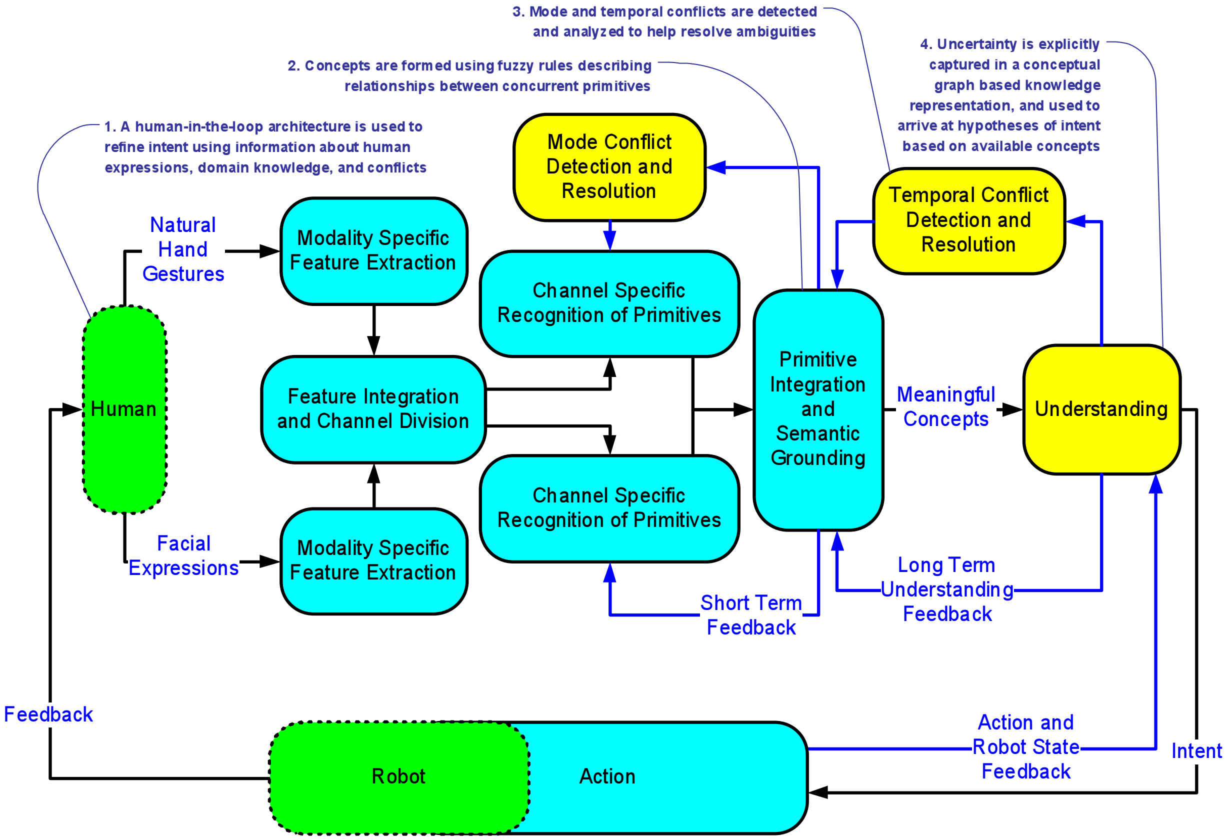

3. Mode and temporal conflicts are detected and analyzed to help resolve ambiguities
2. Concepts are formed using fuzzy rules describing relationships between concurrent primitives
4. Uncertainty is explicitly captured in a conceptual graph based knowledge representation, and used to arrive at hypotheses of intent based on available concepts
1. A human-in-the-loop architecture is used to refine intent using information about human expressions, domain knowledge, and conflicts
Mode Conflict Detection and Resolution
Temporal Conflict Detection and Resolution
Natural Hand Gestures
Modality Specific Feature Extraction
Channel Specific Recognition of Primitives
Human
Feature Integration and Channel Division
Primitive Integration and Semantic Grounding
Meaningful Concepts
Understanding
Facial Expressions
Modality Specific Feature Extraction
Channel Specific Recognition of Primitives
Short Term Feedback
Long Term Understanding Feedback
Feedback
Action and Robot State Feedback
Intent
Robot
Action

Figure 7.1: The presented approach leverages domain specific knowledge to identify reasonable hypotheses of intent from observations of human expression. Primitives are recognized from basic facial expressions and hand gestures and combined to form meaningful concepts. Temporal conflicts, mode conflicts, and domain specific knowledge are used to refine hypotheses of human intent.

ognizing expressions using a small set of low-level concurrent primitives. Using a small set of primitives simplifies the recognition process, in addition to providing a language independent base that can be used to describe diverse forms of human expression. Primitives corresponding to basic facial expressions and hand gestures are modelled and recognized using a FHMM for each dynamic primitive, and a FIS to leverage human descriptions of static primitives. Although the recognition performance between the FHMM and HMM are comparable for the small set of defined primitives, the FHMM approach typically required fewer iterations to converge during training. Relying on machine learning approaches for these relatively simple primitives rather than extending the models to represent longer, more complex forms of expression is intentional in this design. The emphasis is shifted away from conventional formal syntactical analysis toward a stronger reliance on human maintainable concepts and domain knowledge. Focusing on a small set of known primitives, and leveraging human knowledge for higher level processing, is appropriate in a domestic service robot application, as its range of capabilities must be clearly defined regardless of the user interface. The knowledge base used to define a domestic service robot's capabilities can be shared and augmented to support the multimodal interface while avoiding some of the sparse data problems that occur when relying on machine learning techniques to train large, complex models of language.

Sets of concurrent primitives from all sources are brought together and used to form meaningful concepts that explain the observed primitives. Human knowledge is leveraged in the form of fuzzy temporal constraints to describe the relationship and relevance of each primitives with respect to each concept. This approach allows meaningful but vague human knowledge about concepts to enhance the precise primitives derived from perceiving human expression. This approach allows concept definitions to be easily maintained in a human readable manner, while providing flexibility to express a concept in terms of any combination of primitives.

At a larger time scale, sets of concepts are evaluated against domain specific knowledge in the form of conceptual graphs to arrive at hypotheses of intent. Since knowledge bases can potentially be very large, yet insufficient to explain all possible scenarios, approximations are made to find reasonable but not necessarily exact matches. An analogical reasoning approach is used to accommodate the imprecision and inherent variations between human expressions of the same underlying intent. The structure and relevance of each concept in a conceptual graph in addition to the similarity between concepts is used to determine similarity. Information about the similarity is used to search for the most likely underlying intent, using standard conceptual graph operations in a Tabu search algorithm to explore alternatives.

The quality of the resulting hypothesis of intent is further improved by leveraging conflicts and providing feedback to refine both the underlying concepts and basic concurrent primitives. This process of actively searching for conflicts is important, as ignorance of conflicts can result in an incorrect hypothesis of intent based on a faulty source of information. Conflicts between primitives that cannot co-exist in the same channel are identified as mode conflicts and used to provide feedback and help refine the bounded set of primitives used to form concepts. Similarly, temporal conflicts between concepts that cannot co-exist at the same time are used in combination with the hypotheses of intent to help refine the bounded set of concept candidates. By embracing conflict and feedback as integral components of the framework, the average accuracy of individual concepts and the resulting hypothesis of intent is improved over a typical open-loop approach. This improved accuracy is important in reducing human time and effort when compared against an open-loop approach, or an approach that naïvely pre-selects a primary modality to resolve conflicts without considering context. Any improvement in accuracy reduces the need for a human participant to repeat their communication, and reduces the need to exert additional effort to express their intentions in an alternate manner.

This architecture leverages existing human expertise in a maintainable manner at the highest level, including in the knowledge base and concept definitions, and uses machine trained concurrent primitives as the basis of this knowledge. The concurrent nature of the primitives allows a small set of trained primitives to express a wide range of concepts, whereas the human maintainability of the concepts and knowledge base is important to describe the flexible scenarios in which domestic service robots operate. The ability for a robot or other intelligent device to understand human expression is critical to save valuable human time and effort and essential to realize a transparent or ubiquitous computing environment.

7.1 Publications

This research, and relevant related work, resulted in several journal publications, including:

- W.B. Miners and O.A. Basir. Toward Understanding Expression for Tele-Operation. *International Journal of Computer Applications in Technology: Special Issue on Collaborative Multimedia Applications in Technology*, (Accepted), 2006.
- W.B. Miners, O.A. Basir, and F. Karray. Intelligent Object Feature Identification Using A Mobile Sonar Rangefinder. *International Journal of Intelligent Automation and Soft Computing*, (In Press), 2006.
- W.B. Miners, O.A. Basir, and M.S. Kamel. Understanding Hand Gestures Using Approximate Graph Matching. *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, Vol. 35, No.2, pages 239–248, March 2005.

Some of the related conference publications include:

- A. Tehrani, B. Gruneir, B. Miners, A. Khamis, H. Li, M. El-Abd, I. Song, M. Kamel, O. Basir, and F. Karray. An Agent-Based Architecture for a Multi-Robot System, *International Conference on Automation, Robotics, and Autonomous Systems*, Cairo, Egypt, December 19-21 2005.
- W.B. Miners and O.A. Basir, Dynamic Facial Expression Recognition Using Fuzzy Hidden Markov Models. *IEEE International Conference on Systems, Man, and Cybernetics*, 2005.
- W.B. Miners, O.A. Basir, and M.S. Kamel. Knowledge-based Disambiguation of Hand Gestures. In *Proceedings of the 2002 IEEE International Conference on Systems, Man, and Cybernetics*, pages 198–203, October 2002.
- W.B. Miners and O.A. Basir. Real-Time Issues in On-Chip Hand Gesture Recognition. In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics - Industrial Systems*, pages 496–501, 2001.

Numerous additional publications are either currently in progress, or in the planning stages to disseminate relevant research activities in areas including:

- Recognition without correspondence,
- Knowledge-based object anchoring with low-complexity sensors,
- Leveraging conflicts in human-robot interaction,
- Lightweight frameworks for multi-agent communication, and
- Applications of conceptual structures in multimodal communication.

7.2 Future Directions

A plethora of opportunities exist to further explore both parameters of the architecture itself, and individual subsystems within the presented architecture. Algorithms applied for perception and recognition can be altered or replaced to evaluate the impact on the overall system and resulting performance. The flexible nature of the presented architecture can be adapted and evaluated using any number of alternative sources of information, including conventional sources such as speech, whole-body expression, and handwriting. Incorporating less conventional sources including odour, thermal emissions, and bio-electric activity also provides opportunities to explore the relationships between these often neglected forms of expression and ones underlying intent.

Since the emphasis of this research is on the understanding of human expression in multimodal natural human-robot communication, opportunities exist to experiment with alternate preprocessing and feature extraction methods within the same overall framework. One of the drawbacks to using a histogram based approach for facial features is its user-dependence. It would be valuable to explore or integrate alternate algorithms to provide user-independent features for facial expressions. Other opportunities exist on the task planning and robot control side, where alternative environments, devices, and control algorithms utilizing the knowledge-based framework can be evaluated.

A major aspect of human-robot interaction that is critical in almost every implementation, and complements this work is the robot-human aspect. Since the focus of this research is on understanding human expressions in human-robot communication, little emphasis was placed on information flowing in the reverse direction. Nevertheless, effective communication between a personal service robot and a person is vital in many scenarios, and presents a whole new set of challenges, including bidirectional interactivity. Relevant challenges also include effectively handling conversations, where the intent of the person is not a directive but a query, actively soliciting the attention of a human, and physically interacting with people in real-world domestic environments. Interest in human-robot interaction is continually growing, as is the need for both efficient human-robot and robot-human interfaces.

Before diverging too far from the focus of this research, it is important to mention the opportunities to delve even deeper into several specific aspects of the presented architecture. These opportunities include exploring alternate topologies to more accurately model dynamic human hand gestures and facial expressions. Left-right HMM topologies are commonly used due to their simplicity, but it would be a valuable exercise to evaluate the performance of more complex models. Optimization with both large scale knowledge bases, and with user-independent datasets of multimodal human expression is also very important to apply this approach outside the laboratory, as is evaluating online continuous implementations. Additional relevant areas that will be of value include exploring the balance between human adaptation and machine learning to optimize the training and performance of not just the device, but also the human participant. The remainder of this chapter briefly discusses relevant future directions based on questions and issues that were encountered during this research.

Context, Speech, and Other Useful Information Sources

Although the performance of the presented architecuture was discussed in terms of facial expressions and hand gestures, the architecture is not limited to these two sources of information. Obtaining useful information from hand gestures and facial expressions can improve the accuracy of existing speech recognition systems by providing complementary and redundant information about human intent. Similarly, speech, lip movement, and other sources of human expression can be integrated into this architecture at the conceptual level to improve the overall robustness and accuracy of understood intent. It will be especially interesting to compare closely correlated sources of information such as lip movement and speech against loosely correlated sources such as hand gestures and facial expressions.

Another valuable aspect of the conceptual level approach in this architecture is the opportunity to incorporate not only other forms of human expression, but also context as a source of information. Context may include relevant information about the current activity of the robot or device, information about the current environment, or even historical information about past interactions. The most appropriate location to incorporate context is after concepts are formed from human expressions, to help guide the search through domain specific knowledge. Historical information can be used to focus the search on related scenarios, whereas information about the current environment and activity of the robot or device can be used to limit the scope of the search.

Handling Parameterized Expressions

Some expressions may contain the same movements or positions, but convey different meaning based on specific variations in rate or dynamics. A simple example is the difference between a gentle arm movement, and a forceful arm movement. Both movements can follow the same path, but differ in velocity and emphasis. The HMM based approach used to recognize primitives was selected in part due to its ability to compensate for timing variations. Unfortunately, this ability to compensate for timing variations makes it difficult to preserve variations in dynamics during the recognition process. A gesture that has different meaning when performed at different rates can be distinguished by treating each different rate or emphasis as a separate primitive. As long as a set of discrete rates can be identified, it should be possible to train and use these separate primitives. Alternate approaches to augment primitives should be considered if continuous information is desired, or if the recognition performance of separate primitives is inadequate. This may include augmenting primitives with rate or direction information that, while not directly used for recognition, can be useful when forming concepts and in higher-level processing.

User Independence

Although challenges exist when moving from a user-dependent to a user-independent system, in this architecture, they are isolated within the recognition of primitives. Perhaps the largest challenge is building or finding a suitable existing dataset large enough to adequately represent variations between primitives when performed by any individual. Primitives are currently represented using a single model, but this approach may have to change when moving to a user-independent system. Multiple models can be used to help represent the same primitive when significant variations exist across different users. No modifications are required to form concepts, understand hypotheses of intent, or leverage conflicts when incorporating user-independent primitives.

Multimodal Fission and Cross-modal Attention

Multimodal fusion research tackles the problem of handling information input from from multiple modes and modalities into a machine. The reverse situation of generating information across multiple modes and modalities is a research area that shares several common issues with multimodal fusion. One of the major differences between generation and processing of multimodal information is the consideration of human limitations. When processing multimodal information, we are analyzing human expression by observation. When generating this information, however, decisions have to be made to balance the use of multiple modalities against the ability of us to handle this information. Research suggests that our ability to understand information conveyed across different modalities does not necessarily improve by including additional modalities. Although this research area lies in the realm of cognitive psychology, a good understanding of how humans divide their attention amongst available senses will help to design systems that efficiently convey information from a robot back to a person regardless of the field of research.

Mixed-Initiative Systems

The closed-loop approach to human-robot interaction can be advanced in future work by incorporating mixed-initiative research. Mixed-initiative systems take advantage of human strengths and device strengths where appropriate, rather than attempting to blindly automate everything. This approach offers practical solutions to allow a robot to carry out tasks mostly on their own with the support of human input to help with difficult problems. A compromise is made between required human communication effort and the ability for a robot to accomplish its goal.

Attention and Multiple Participants

This thesis assumes that the person communicating intent is communicating directly with a single robot. In environments with multiple people, robots, and without the direct communication assumption, the attention or focus of the person must be determined. Identifying the focus of communication can be challenging when the person is not gazing directly at their target, as is often the case when communication involves deictic hand gestures or explanations by example.

Expression versus Intent

Unfortunately, there can be a large disparity in practice between human expression and underlying intent. In today's society, humans can often generate a false expression to mask their true intent. It will be valuable (and challenging) to investigate approaches to accurately detect and compensate for false expressions. It should be possible to extend the architecture presented in this thesis to include observations of direct brain emissions, and other forms of human expression that are difficult to mask. This additional information can be used to detect and penalize false expressions in an attempt to determine the true underlying intent.

Online Refinement of Primitives and Concepts

In many situations, a static domain specific knowledge base combined with a comprehensive predefined set of language primitives will provide an adequate level of support for human-robot interaction. However, as the scope of capabilities of individual robots and devices expand, it becomes increasingly difficult to define the necessary domain specific knowledge to encompass all reasonable interaction scenarios. It will be valuable for a robot to dynamically adapt and expand its knowledge base as it experiences successful and unsuccessful interactions. Although effective online refinement of knowledge is a major ongoing challenge in the artificial intelligence community [129], it will help shift some of the knowledge base development work traditionally performed manually to the robot itself.

Using Intent To Carry Out an Action

Performing an action or providing a response is required to provide feedback from the robot to the human participant. Although this feedback component does not directly involve multimodal integration, it is still an important aspect of a real-world implementation. This feedback is required to acknowledge that human communication was either understood completely, partially understood, or not understood at all. Without feedback, there is no way for the human participant to know that their intent was communicated successfully. Of course, it is possible on conventional PCs, or in a lab environment, to provide feedback in the form of a simple on screen message. Unfortunately, many devices and robots do not have the luxury of a monitor, and often require some form of physical interaction with the real world to successfully complete a task.

Anchoring

The ability to understand meaningful concepts has limited use in the real-world until these concepts are anchored or associated with the physical environment. This process of anchoring is a rapidly developing research area [130] and is required to ensure the concepts we express are properly associated with concepts in the robot's environment (Figure 7.2). In the presented architecture, it is logical to share the same knowledge-base used for understanding human expression with an action component. This common knowledge-base can tie the logic that defines an instance of concept and its relationships together with physical attributes of an object that can be used for perception. Transient information about object instances can then be easily shared, including the location of one instance of an object. Sharing this transient information allows future research to resolve ambiguous deictic hand gestures, replacing a trajectory with a more meaningful concept or object instance.

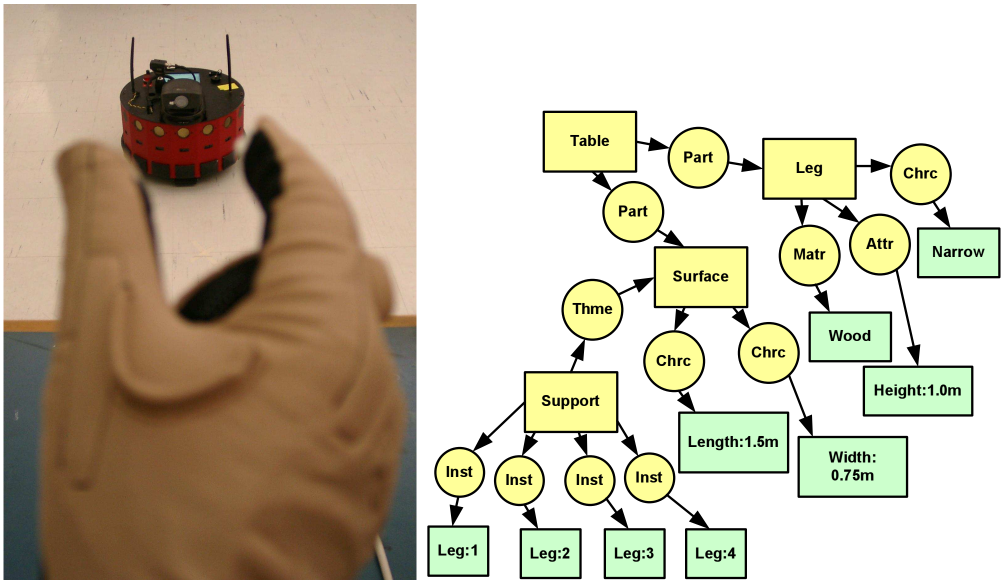


Figure 7.2: The concepts understood from human expression can be anchored to concepts about objects in the robot's environment using a shared knowledge-base between the action and understanding components of the architecture.

Perception

Chapter 3 focused on perceiving human expression, however, perception of the remainder of the environment is also important to successfully carry out actions. In a domestic environment, it is reasonable to assume that a map of the permanent objects in the environment in which the robot navigates is known *a priori*. This simplification can be justified as the layout of fixed walls, doors, and other permanent objects in a home or building in which a domestic service robot operates rarely change. On the other hand, many approaches exist for the popular simultaneous robot localization and mapping problem to determine both the environment and the robot's location in that environment. In addition to walls, doors and other permanent objects, transient objects must also be detected and tracked within the known environment to carry out most practical tasks. During this research, the author explored an approach to identify objects using low-complexity sonar sensors rather than the common vision approach. Planar features and corners were identified using a Hough transform approach, and objects detected from these features using a fuzzy rule-based inference system [131].

The Hough transform approach transfers sonar observations from world space (x, y) to Hough space (ρ, θ) by representing possible lines that pass through the point (x, y) in polar form using $\rho = xcos\theta + ysin\theta$. The length ρ and angle θ of the normal to each line that can pass through the point (x, y) is used to increment an accumulator for a line existing at (ρ, θ). Local maxima identify planar features and points. This low-complexity approach is applied incrementally after short fixed movement distances as proposed for some mapping tasks [132].

Task Planning

When a meaningful interpretation of human expression is available, a sequence of tasks must be identified to carry out the desired intent. These tasks could include an element of time together with some combination of:

1. Navigating to a specific location in the known environment
2. Moving an object between two locations
3. Locating a known object in an unknown location
4. Providing simple human feedback

A wide range of domestic service robot tasks can be represented using a combination of these four basic tasks, including delivery of medication or difficult to reach items, simple meal preparation, and providing reminders. The same small set of robot tasks can also describe the primary functionality of conventional robotic vacuum cleaners and lawn-mowers, even though their tasks are highly constrained.

Developing planners for complex tasks is an ongoing challenge in the robotics and artificial intelligence fields. Classical approaches to planning tackle the planning problem as a search for a sequence of actions to reach a goal state given the initial state. The use of the four previously defined basic tasks helps to keep planning in this specific domain simple. Unfortunately, even with a restricted planning domain, classical planners are intractable on most real-world problems. These include both the popular Graphplan based approaches [133] and estimated-regression approaches [134]. To realize an online implementation of this system in future work, and take advantage of the existing shared knowledge, a case based planner (CBP) is recommended [135]. Although CBP approaches are not new, and are sometimes discounted due to the need for expert knowledge, CBP approaches can be feasibly applied online to practical real-world problems.

A case-based planner adapts similar prior examples to current planning problems. A planner can take advantage of the same analogical reasoning framework we used to combine concepts and understand human expression. Past examples of tasks and their associated actions can be stored in the form of conceptual graphs. The shared knowledge base can then be used together with these past examples to find analogies between desired tasks and prior experience.

Bibliography

[1] A. Mehrabian. Communication without words. *Psychology Today*, 2(4):53–56, 1968.

[2] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.

[3] S. L. Oviatt. Breaking the robustness barrier: Recent progress on the design of robust multimodal systems. In M. Zelkowitz, editor, *Advances in Computers*, volume 56, pages 305–341. Academic Press, 2002.

[4] J.-L. Nespoulous, P. Perron, and A. R. Lecours. *The Biological Foundations of Gestures: Motor and Semiotic Aspects*. Lawrence, Erlbaum Associates, N.J., 1986.

[5] A. Kendon. Current issues in the study of gestures. In *The Biological Foundations of Gestures: Motor and Semiotic Aspects*, pages 23–47. Lawrence, Erlbaum Associates, New Jersey, 1986.

[6] D. McNeill. *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago Press, Chicago, 1992.

[7] J. Cassell and D. McNeill. Non-verbal imagery and the poetics of prose. *Poetics Today*, 12(3):375–404, 1991.

[8] G. Kurtenbach and E. Hulteen. Gestures in human-computer communications. In B. Laurel, editor, *The ARt of Human Computer Interface Design*, pages 309–317. Addision-Wesley, 1990.

[9] P. Eisert and B. Girod. Analysing facial expressions for virtual conferencing. *IEEE Transactions on Computer Graphics and Applications*, 18(5):70–78, 1998.

[10] L. Nigay and J. Coutaz. A design space for multimodal systems: concurrent processing and data fusion. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 172–178. ACM Press, 1993.

[11] J. Coutaz. Multimedia and multimodal user interfaces: A taxonomy for software engineering research issues. In *Proceedings of the Second East-West HCI conference*, pages 229–240, August 1992.

[12] W. Freeman and M. Roth. Orientation histogram for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, pages 296–300, Zurich, 1995.

[13] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[14] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.

[15] T. J. Darrell, I. A. Essa, and A. P. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1236–1242, December 1996.

[16] J. Triesch and C. von der Malsburg. Robust classification of hand postures against complex backgrounds. In *Proc. of the 2nd International Conference on Automatic Face and Gesture Recognition*, pages 170–175. IEEE Computer Society Press, 1996.

[17] J. Triesch and C. Malsburg. Robotic gesture recognition. In *Gesture Workshop*, pages 233–244, Bielefeld, Germany, 1997.

[18] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, July 1997.

[19] Y. Yacoob and L. Davis. Computing spatio-temporal representations of human faces. In *CVPR94*, pages 70–75, 1994.

[20] P. Ekman and W. Friesen. *Facial Action and Coding System.* Consulting Psychologists Press, Palo Alto, USA, 1978.

[21] A. Kapoor. *Automatic Facial Action Analysis.* PhD thesis, MIT, Massachusetts, June 2002.

[22] S. Kawato and N. Tetsutani. Real-time detection of between-the-eyes with a circle frequency filter. In *ACCV*, pages 442–447, 2002.

[23] Y. Cui, D. L. Swets, and J. Weng. Learning-based hand sign recognition using SHOSLIF-m. In *IEEE International Conference on Computer Vision*, pages 631–636, 1995.

[24] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *IEEE International Conference on Computer Vision*, pages 786–793, 1995.

[25] R. Hamdan, F. Heitz, and L. Throaval. Gesture localization and recognition using probabilistic visual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 98–103, 1999.

[26] Y. Iwai, H. Shimizu, and M. Yachida. Real-time context-based gesture recognition using hmm and automaton. In *Proceedings of International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 127–134, Corfu, Greece, September 1999.

[27] A. Wilson, A. Bobick, and J. Cassell. Recovering the temporal structure of natural gesture. In *Proc. of the International Conference on Automatic Face and Gesture Recognition*, pages 66–71, Killington, Vermont, 1996.

[28] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, June 1994.

[29] A. Yilmaz, K. Shafique, and M. Shah. Estimation of rigid and non-rigid facial motion using anatomical face model. In *International Conference on Pattern Recognition*, volume 1, pages 377–380, Quebec, Canada, 2002.

[30] H. Ouhaddi and P. Horain. 3D hand gesture tracking by model registration. In *Proceedings of the International Workshop on Synthetic - Natural Hybrid Coding and Three Dimensional Imaging*, pages 70–73, Santorini, Greece, September 1999.

[31] R. Rosales, V. Athitsos, and S. Sclaroff. 3D hand pose reconstruction using specialized mappings. In *IEEE International Conference on Computer Vision*, pages 378–385, Vancouver, BC, Canada, 2001.

[32] J. Segen and S. Kumar. Shadow gestures: 3D hand pose estimation using a single camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 479–485, 1999.

[33] M. Agus, F. Bettio, E. Gobbetti, and L. Fadiga. An integrated environment for steroscopic acquisition, off-line 3D elaboration, and visual presentation of biological actions. In J. D. Westwood, H. M. Hoffmann, G. T. Mogel, D. Stredney, and R. A. Robb, editors, *Medicine Meets Virtual Reality 2001 – Inner Space, Outer Space, Virtual Space*, pages 23–29, Amsterdam, The Netherlands, January 2001. IOS.

[34] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with physical forces. *Computer Vision and Image Understanding*, 81:328–357, 2001.

[35] T. Ishikawa, S. Morishima, and D. Terzopoulos. 3D face expression estimation and generation from 2D image based on a physically constrained model. *IEICE Transactions on Information and Systems*, E83-D(2), February 2000.

[36] M. Pitermann and K. G. Munhall. An inverse dynamics approach to face animation. *Journal of the Acoustical Society of America*, 110(3):1570–1580, September 2001.

[37] J. Scheirer, R. Fernandez, and R. W. Picard. Expression glasses: A wearable device for facial expression recognition. In *CHI*, Short Papers, Pittsburgh, PA, USA, 1999.

[38] S. Lu, S. Igi, H. Matsuo, and Y. Nagashima. Towards a dialogue system based on recognition and synthesis of japanese sign language. In *International Gesture Workshop Proceedings*, pages 259–271, 1997.

[39] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden Markov models. In *Proc. of IEEE International Symposium on Computer Visison*, volume 29, pages 265–270, 1995.

[40] G. S. Schmidt and D. H. House. Towards model-based gesture recognition. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 416–421, Grenoble, France, March 2000.

[41] R.-H. Liang and M. Ouhyoung. Real time continuous gesture recognition system for sign language. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pages 558–565, 1998.

[42] C. Vogler and D. Metaxas. Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods. In *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 156–161, 1997.

[43] A. Wilson and A. Bobick. Realtime online adaptive gesture recognition. In *Proceedings of the International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real Time Systems*, pages 270–275, 1999.

[44] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[45] C. Vogler and D. Metaxas. Parallel hidden Markov models for american sign language recognition. In *IEEE International Conference on Computer Vision*, pages 116–122, 1999.

[46] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[47] C. Lee and Y. Xu. Online, interactive learning of gestures for human/robot interfaces. In *IEEE Int. Conf. on Robotics and Automation*, pages 2982–2987, 1996.

[48] A. Wilson and A. Bobick. Recognition and interpretation of parametric gesture. In *IEEE International Conference on Computer Vision*, pages 329–336, Bombay, India, January 1998.

[49] P. Hong, M. Turk, and T. S. Huang. Gesture modeling and recognition using finite state machines. In *Proceedings of the IEEE Conference on Face and Gesture Recognition*, pages 410–415, Grenoble, France, March 2000.

[50] M. Johnston and S. Bangalore. Finite-state multimodal parsing and understanding. In *International Conference on Computational Linguistics*, pages 369–375, 2000.

[51] M. Yang and N. Ahuja. Recognizing hand gesture using motion trajectories. In *CVPR*, pages 466–472, 1999.

[52] C. Nolker and H. Ritter. Detection of fingertips in human hand movement sequences. In *International Gesture Workshop Proceedings*, pages 209–218, 1997.

[53] H. Boehme, A. Brakensiek, U. Braumann, M. Krabbes, and H. Gross. Visually-based human-machine interaction in a neural architecture. In *International Gesture Workshop*, pages 219–232, 1997.

[54] E. Littmann, A. Drees, and H. Ritter. Visual gesture-based robot guidance with a modular neural system. *Advances in Neural Information Processing Systems*, 8:903–909, 1996.

[55] S. S. Fels and G. E. Hinton. Glove-Talk: A neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks*, 4(1):2–8, January 1993.

[56] J. Kramer and L. Liefer. The ’talking-glove’: A speaking aid for nonvocal deaf and deaf-blind individuals. In *Proceedings of RESNA 12th Annual Conference*, pages 471–472, Lousiana, 1989.

[57] R. Bolt. Put-that-there: Voice and gesture at the graphics interface. *Computer Graphics*, 14:262–270, 1980.

[58] N. Krahnstoever, S. Kettebekov, M. Yeasin, and R. Sharma. A real-time framework for natural multimodal interaction with large screen displays. In *Proceedings of the Fourth International Conference on Multimodal Interfaces*, October 2002.

[59] W. Wahlster. Smartkom: Fusion and fission of speech, gestures, and facial expressions. In *Proceedings of the 1st International Workshop on Man-Machine Symbiotic Systems*, pages 213–225, Kyoto, Japan, 2002.

[60] R. Bischoff and V. Graefe. Dependable multimodal communication and interaction with robotic assistants. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, pages 300–305, Berlin, 2002.

[61] L. Chen, T. Huang, T. Miyasato, and R. Nakatsu. Multimodal human emotion / expression recognition. In *International Conference on Automatic Face and Gesture Recognition*, pages 366–371, 1998.

[62] L. D. Silva, T. Miyasato, and R. Nakatsu. Facial emotion recognition using multimodal information. In *Information, Communication and Signal Processing Conference*, pages 397–401, 1997.

[63] S. Kettebekov and R. Sharma. Toward natural gesture/speech control of a large display. In *IFIP Working Conference on Engineering for Human Computer Interaction*, pages 221–234, 2001.

[64] F. Quek. The catchment feature model: A device for multimodal fusion and a bridge between signal and sense. *EURASIP Journal of Applied Signal Processing*, 11:1619–1636, 2004.

[65] A. Ross and A. K. Jain. Information fusion in biometrics. *Pattern Recognition Letters, Vol. 24, Issue 13, pp. 2115-2125*, 24:2115–2125, Sep 2003.

[66] S. L. Oviatt and P. R. Cohen. Multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3):45–53, 2000.

[67] F. Quek. The catchment feature model for multimodal language analysis. In *Proceedings of the Ninth IEEE International Conference on Computer Vision and Pattern Recognition*, pages 540–547, October 2003.

[68] A. Bobick and Y. Ivanov. Action recognition using probabilistic parsing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 196–202, Santa Barbara, California, 1998.

[69] R.-H. Liang and M. Ouhyoung. A sign language recognition system using hidden Markov model and context sensistive search. In *Proceedings of the ACM Symposium on Virtual REality Software and Tehcnology*, pages 59–66, June 1996.

[70] A. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. *Royal Society Workshop on Knowledge-based Vision in Man and Machine*, pages 1257–1265, February 1997.

[71] C. Hummels and P. Stappers. Meaningful gestures for human computer interaction: Beyond hand postures. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 591–596, Nara, Japan, April 1998.

[72] C. S. Pinhanez and A. F. Bobick. Human action detection using PNF propagation of temporal constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 898–904, 1998.

[73] R. C. Schank. *Conceptual Information Processing.* North-Holland, Amsterdam, 1975.

[74] M. Billinghurst, J. Savage-Carmona, P. Oppenheimer, and C. Edmond. The expert surgical assistant: An intelligent virtual environment with multimodal input. In S. Weghorst, H. Sieberg, and K. Morgan, editors, *Medicine Meets Virtual Reality IV*, pages 590–607, San Diego, CA, 1995. IOS Press.

[75] J. F. Sowa. *Conceptual Structures: Information Procesing in mind and Machine.* Addison-Wesley, Massachusetts, 1984.

[76] W. Freeman and C. Weissman. Television control by hand gestures. In M. Bichsel, editor, *International Workshop on Automatic Face and Gesture Recognition*, pages 179–183, Zurich, Swizerland, 1995.

[77] T. W. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4):143–166, 2003.

[78] M. Brand and I. Essa. Casual analysis for visual gesture understanding. In *AAAI Fall Symposium on Computational Models for Integrating Language and Vision*, 1995. Also available as Vision and Modeling Technial Report 327, MIT.

[79] R. Mayer. Cognition and instruction: Their historic meeting within educational psychology. *Journal of Educational Psychology*, 84(4):405–412, 1992.

[80] C. Theobalt, J. Bos, T. Chapman, A. Espinosa-Romero, M. Fraser, G. Hayes, E. Klein, T. Oka, and R. Reeve. Talking to godot: Dialogue with a mobile robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, pages 1338–1343, 2002.

[81] L. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, and R. Dillmann. Using gesture and speech control for commanding a robot assistant. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, pages 454–459, Berlin, Germany, September 2002. IEEE Press.

[82] S. Iba, J. J. Paredis, and P. K. Khosla. Interactive multi-modal robot programming. In *Proceedings of the International Conference on Robotics and Automations (ICRA)*, pages 161–168, Washington, D.C., May 2002.

[83] A. Corradini, M. Mehta, N. Bernsen, and J. Martin. Multimodal input fusion in human-computer interaction on the example of the on-going nice project. In *Proceedings of the NATO-ASI conference on Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*, Yerevan (Armenia), August 2003.

[84] S. Basu. *Conversational Scene Analysis*. PhD thesis, Massachusetts Institute of Technology, USA, 2002.

[85] W. B. Miners. Hand gesture understanding for interactive service robots. Master's thesis, University of Guelph, 2002.

[86] V. Athitsos and S. Sclaroff. An appearance-based framework for 3D hand shape classification and camera viewpoint estimation. In *IEEE Automatic Face and Gesture Recognition Conference*, pages 45–52, Washington, D.C., May 2002.

[87] D. G. Kamper, E. G. Cruz, and M. P. Siegel. Stereotypical fingertip trajectories during grasp. *Journal of Neurophysiology*, 90(6):3702–3710, September 2003.

[88] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003.

[89] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *IEEE International Conference on Computer Vision (ICCV)*, pages 734–741, October 2003.

[90] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-based modeling in particle filters. *IEEE Transactions on Image Processing*, 13:1491–1506, November 2004.

[91] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *Proceedings of the 7th International Conference on Computer vision*, pages 1165–1173, Kerkya, Greece, 1999.

[92] G. R. Badski. Real time face and object tracking as a component of a perceptual user interface. In *4th IEEE Workshop on Applications of Computer Vision*, Princeton, New Jersey, 1998.

[93] B. Moghaddam, J. Lee, H. Pfister, and R. Machiraju. Model-based 3D face capture with shape-from-silhouettes. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*, pages 20–27, October 2003.

[94] C. Theobalt, M. A. Magnor, P. Schuler, and H.-P. Seidel. Combining 2D feature tracking and volume reconstruction for online video-based human motion capture. *International Journal of Image and Graphics*, 4(4):563–583, 2004.

[95] F. Bourel, C. C. Chibelushi, and A. A. Low. Robust facial expression recognition using a state-based model of spatially-localised facial dynamics. In *The 5th International Conference on Automatic Face and Gesture Recognition*, pages 113–118, Washington D.C., USA, May 20-21 2002.

[96] R. S. Feris, J. Gemmell, K. Toyama, and V. Krueger. Hierarchical wavelet networks for facial feature localization. In *The 5th International Conference on Automatic Face and Gesture Recognition*, pages 125–130, Washington D.C., USA, May 20-21 2002.

[97] M. Mohamed and P. Gader. Generalized hidden Markov models. i. theoretical frameworks. *IEEE Transactions on Fuzzy Systems*, 8(1):67–81, February 2000.

[98] M. Mohamed and P. Gader. Generalized hidden Markov models. ii. application to handwritten word recognition. *IEEE Transactions on Fuzzy Systems*, 8(1):82–94, February 2000.

[99] *Hamburg Notation System for sign language. Development of sign writing with a computer application.*, Hamburg, July 1989. Signum-Verlag.

[100] W. C. Stokoe, D. Casterline, and C. Croneberg. *A Dictionary of American Sign Language on Linguistic Principles.* Linstok Press, Silver Spring, 1965.

[101] V. Sutton. A way to analyse american sign language (ASL) and any other sign language without translation into any spoken language. In F. Caccamise, M. Garretson, and U. Bellugi, editors, *Teaching American Sign Language as a second language. Proceedings of the Third National Symposium on Sign Language Research and Teaching*, pages 204–213, Boston, MA, October 1982.

[102] R. Elliott, J. R. W. Glauert, V. Jennings, and J. R. Kennaway. Sigml notation and sigml-signing software system. In *Workshop on the Representation and Processing of Sign Languages, Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 98–104, Lisbon, Portugal, May 2004.

[103] L. R. Rabiner and B. Juang. An introduction to hidden Markov models. *IEEE Transactions on Acoustics Speech, Signal Processing*, 3(1):4–16, 1986.

[104] S. Iba, J. V. Weghe, C. Paredis, and P. Khosla. An architecture for gesture based control of mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99)*, volume 2, pages 851–857, October 1999.

[105] B. Bauer and K.-F. Kraiss. Towards a 3rd generation mobile telecommunication for deaf people. In *10th Aachen Symposium on Signal Theory*, pages 101–106. VDE Verlag, 2001.

[106] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[107] T. H. Cao. A formalism for representing and reasoning with linguistic information. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(3):281–307, 2002.

[108] R. Thomopoulos, P. Buche, and O. Haemmerle. Different kinds of comparisons between fuzzy conceptual graphs. In *Proceedings of the 11th International Conference on Conceptual Structures*, pages 54–68, 2003.

[109] D. B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[110] J. F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundataions*. Brooks/Cole, 2000.

[111] J. F. Sowa and A. K. Majumdar. Analogical reasoning. In *Proceedings of the 11th International Conference on Conceptual Structures*, pages 16–38, 2003.

[112] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NPCompleteness*. W.H. Freeman and Company, New York, 1979.

[113] C. Ware and R. Balakrishnan. Reaching for objects in VR displays: Lag and frame rate. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(4):331–356, 1994.

[114] B. Watson, N. Walker, W. Ribarsky, and V. Spaulding. The effects of variation of system responsiveness on user performance in virtual environments. *Human Factors: Special Issue on Virtual Environments*, 3(40):403–414, 1998.

[115] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operation Research*, 5:533–549, 1986.

[116] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, pages 746–748, December 1976.

[117] D. Ballou and H. Pazer. Modeling completeness versus consistency tradeoffs in information decision contexts. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):240–243, February 2003.

[118] G. Greco, S. Greco, and E. Zumpano. A logical framework for querying and repairing inconsistent databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1389–1408, December 2003.

[119] T. Hazen. Visual model structures and syncrony constraints for audio-visual speech recognition. *IEEE Transactions on Speech and Audio Procesing*, page (To appear), 2006.

[120] J. Carlson and R. R. Murphy. Use of dempster-shafer conflict metric to adapt sensor allocation to unknown environments. In *American Association for Artificial Intelligence*, 2005.

[121] H. B. Danesh and R. Danesh. Has conflict resolution grown up? toward a developmental model of decision making and conflict resolution. *International Journal of Peace Studies*, 7(1):59–76, 2002.

[122] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality. *Biometrika*, 52(3):591–599, 1965.

[123] P. Royston. A remark on algorithm as 181: The W test for normality. *Applied Statistics*, 44(4):547–551, 1995.

[124] M.-N. Park and J.-Y. Ha. Hmm topology optimization with anti-likelihood criterion for handwriting recognition. In *Sixth IASTED International Conference on Signal and Image Processing*, pages 563–566, August 2004.

[125] A. Beim. A model selection criterion for classification: Application to hmm topology optimization. In *Seventh International Conference on Document Analysis and Recognition*, volume 1, pages 104–108, August 2003.

[126] D. Li, A. Biem, and J. Subrahmonia. Hmm topology optimization for handwriting recognition. In *26th Int. Conference on Acoustics, Speech, and Signal Processing*, 2001.

[127] M. Zimmermann and H. Bunke. Hidden markov model length optimization for handwriting recognition systems. In *Eighth International Workshop on Frontiers in Handwriting Recognition*, page 369, 2002.

[128] B. Gruneir. Multiple agent architecture for a multiple robot system. Master's thesis, Department of Systems Design Engineering, University of Waterloo, 2005.

[129] D. Roy. Learning from multimodal observations. In *IEEE International Conference on Multimedia and Exp (ICME)*, New York, NY, 2000.

[130] S. Coradeschi and A. Saffiotti. An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2-3):85–96, 2003.

[131] B. W. Miners, O. A. Basir, and F. Karray. Intelligent object feature identification using a mobile sonar rangefinder. *International Journal of Intelligent Automation and Soft Computing*, page (To appear), 2006.

[132] J. D. Tardos, J. Neira, P. M. Newman, and J. J. Leonard. Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, 21(4):311–330, April 2002.

[133] A. Blum and M. Furst. Fast planning through planning graph analysis. *Artificial Intelligence*, 90:281–300, 1997.

[134] D. McDermott. Reasoning about autonomous processes in an estimated-regression planner. In *Proc. Int'l. Conf. on Automated Planning and Scheduling*, pages 143–152, 2003.

[135] M. T. Cox, H. Munoz-Avila, and R. Bergmann. Planning in case-based reasoning: Commentary. In I. Watson, editor, *Readings in case-based reasoning*. Morgan Kaufmann, 2003.

[136] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book*. Cambridge University Engineering Department, 2002. Manual for HTK3 software.

[137] W. R. Hamilton. *Elements of Quaternions*, volume I-II. Chelsea Publ. Co., New York, 1969.

Appendices

Appendix A

Hidden Markov Models

Fundamental aspects of hidden Markov models are presented in this appendix. All of this material is based on [97, 103, 136], and is included here solely for the convenience of the reader.

A.1 Traditional HMMs

A traditional HMM can be briefly described as a set of connected states in a Markov chain that probabilistically describe observations. The Markov assumption specifies that the probability of any state is dependent only on the previous state, $P(S_t|S_1, S_2, ..., S_{t-1}) = P(S_t|S_{t-1})$ (where S_t is the state at time t). HMMs are *hidden* since they separate the observations from the states. Each observation, O_t (a vector of size M) is not a direct measurement of the state. In this research, observations include resistive bend sensor derived measurements, magnetic tracker information, and facial expression cues. The states, S_t (a vector of size N), however, represent components of dynamic primitives. In speech research, observations may include derived measurements from an audio signal, and states may represent phonemes or syllables.

Each state in a traditional HMM is stochastically related to an observation by an emission probability, $b_j(k)$ (the probability that state S_j will generate observation k). Similarly, each state is related to dependent states by transition probabilities, a_{ij} (the probability of transitioning from state S_i to state S_j).

Graphically, each state is represented using a circular node as shown in Figure A.1. Dependencies between states are represented using directed arcs, or transitions between each node. The three fundamental problems tackled by HMM approaches are as follows:

1. Given an observation sequence $O_1, O_2, O_3, ..., O_T$, and a model $\lambda = (\mathcal{A}, \mathcal{B}, \pi)$ (where π is the initial state distribution), calculate the probability of the observation given the model

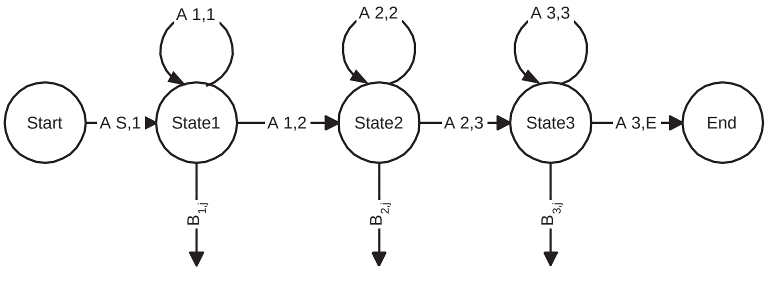


Figure A.1: Hidden Markov Model Parameters

$P(O_1, O_2, ..., O_T|\lambda)$.

2. Given an observation sequence $O_1, O_2, O_3, ..., O_T$, and a model λ, find the optimal state sequence $\text{argmax}_S P(S_1, S_2, S_3, ..., S_T|O_1, O_2, ..., O_T)$ where T is the number of discrete time intervals.

3. Given an observation O_t, estimate the model parameters $\lambda = (\mathcal{A}, \mathcal{B}, \pi)$ to maximize the probability of the observation given the model $P(O_1, O_2, ..., O_T|\lambda)$.

The first of these three problems is determining the model that fits the observation sequence the closest (recognition). The second of these problems is identification of the actual (hidden) state sequence based on given observations, and the third problem is training of the model.

The first problem, calculating $P(\mathcal{O}|\lambda)$ can be calculated using brute force with $O(TN^T)$ complexity as shown in Equation A.1, where $\mathcal{Q}$ is a possible state sequence.

$$\begin{aligned} P(\mathcal{O}|\lambda) &= \sum_{allQ} P(\mathcal{O}, \mathcal{Q}|\lambda) \\ &= \sum_{allQ} \pi_{q_1} b_{q_1}(O_1) \prod_{t=2}^{T} a_{q_{t-1},q_t} b_{q_t}(O_t) \qquad \text{(A.1)} \end{aligned}$$

A more elegant dynamic programming approach can be used instead to determine the probability of the HMM being in state i at time t with complexity $O(TN^2)$. This approach, the forward-backward algorithm, calculates $P(O_1, O_2, ..., O_T|\lambda)$ based on the sum of all possible paths using the forward and backward probabilities. The forward probability $\alpha_t(i)$ is the joint probability of generating the observation sequence $O_1, O_2, ..., O_t$ and being in state i at time interval t. Similarly, the backward probability $\beta_t(i)$ is the joint probability of generating the observation sequence $O_t, O_{t+1}, ..., O_T$ and being in state i at time interval t. The forward $(\alpha_t(i))$ and backward $(\beta_t(i))$ probabilities can be calculated using Equation A.3 and Equation A.5 respectively.

$$\alpha_1(i) = \pi_i b_i(O_1) \qquad \text{(A.2)}$$

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right) b_j(O_t) \qquad \text{(A.3)}$$

$$\beta_T(i) = 1 \qquad \text{(A.4)}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{i,j} b_j(O_{t+1}) \beta_{t+1}(j) \qquad \text{(A.5)}$$

These forward and backward probabilities can be used to calculate $P(O_1, O_2, ..., O_T|\lambda)$ as shown in Equation A.6. The product of the forward and backward probabilities is the joint probability of generating the given observation sequence and arriving at state i at time interval t.

$$P(O_1, O_2, ..., O_T|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(j) \tag{A.6}$$

The second problem is solved using the Viterbi approximation. The Viterbi approximation calculates the likelihood of the best path based on the assumption that only one most likely path exists to reach each state. Rather than summing up the probability of all paths, the Viterbi approximation only uses the most likely path to each state and arrives at its likelihood $\delta_t(i) = \max_{allQ} P(\mathcal{O}, \mathcal{Q}|\lambda)$. The dynamic programming based Viterbi algorithm to calculate this likelihood and the state sequence φ. It is initialized using:

$$\delta_1(i) = \pi_i b_i(O_1) \tag{A.7}$$

$$\varphi_1(i) = 0 \tag{A.8}$$

Iteratively calculated over all $t \geq 2$ and j using:

$$\delta_t(j) = \max_{1\leq i\leq N}(\delta_{t-1}(i)a_{ij})b_j(O_t) \tag{A.9}$$

$$\varphi_t(j) = \underset{1\leq i\leq N}{\operatorname{argmax}}(\delta_{t-1}(i)a_{ij}) \tag{A.10}$$

The state sequence that generates this maximum likelihood result can be calculated using $q_t = \varphi_{t+1}(q_{t+1})$, and the likelihood is given by $P = \max_{1\leq i\leq N} \delta_T(i)$.

The Baum-Welch algorithm is utilized to solve the third problem (training). This iterative algorithm estimates the parameters $\mathcal{A}$ and $\mathcal{B}$ using the forward (α) and backward (β) probabilities. The probability of being in state S_i at time t is represented by $\gamma_t(i)$ and defined in Equation A.11. Similarly, the probability of a path transitioning from state S_i to state S_j between time t and time $t+1$ is represented by $\xi_t(i, j)$ and defined in Equation A.12.

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(\mathcal{O}|\lambda)} \tag{A.11}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(\mathcal{O}|\lambda)} \tag{A.12}$$

New values for a_{ij} and $b_j(k)$ are estimated based on the two probabilities $\gamma_t(i)$ and $\xi_t(i, j)$ The new estimate for a_{ij} is provided in Equation A.14, and the new estimate for $b_j(k)$ is provided in Equation A.15. In Equation A.14, $\sum_{t=1}^{T-1} \xi_t(i, j)$ is the sum of the expected transitions from state S_i to state S_j. Similarly, $\sum_{t=1}^{T-1} \gamma_t(i)$ is the sum of the expected transitions from state S_i to any other state. In Equation A.15, $\sum_{t=1}^{T} \gamma_t(i)$ is the sum of the expected instances of the HMM being in state i, and $\sum_{t=1,O_t=k}^{T} \gamma_t(i)$ is the sum of the expected instances observing k while in state S_i.

$$\pi_i = \gamma_1(i) \tag{A.13}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{A.14}$$

$$b_j(k) = \frac{\sum_{t=1,O_t=k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{A.15}$$

A.2 Generalized Fuzzy HMMs

The generalized fuzzy hidden Markov model consists of fuzzy transition possibilities $\hat{a}_{ij}$, fuzzy observation emission possibilities $\hat{b}_j(k)$, and the initial fuzzy state $\hat{\pi}$. In traditional HMMs, a_{ij} represents the probability of the state at time $t+1$ being S_i given the state at time t is S_j, whereas $\hat{a}_{ij}$ represents the grade of certainty of the state at time $t+1$ being S_i given the state at time t is S_j. Similarly, $\hat{b}_j(k)$ represents the grade of certainty that the generated observation is k given that the state is S_i.

The traditional forward-backward variables from Equation A.3 and Equation A.5 now become Equation A.16 and Equation A.17 when the Choquet integral is used.

$$\hat{\alpha}_{t+1}(j) = \left[\sum_{i=1}^{N} \hat{a}_{ij}\rho_t(i,j)\hat{\alpha}_t(i)\right]\hat{b}_j(O_{t+1}) \tag{A.16}$$

$$\hat{\beta}_t(i) = \sum_{j=1}^{N} \hat{a}_{ij}\rho_t(i,j)\hat{\beta_{t+1}}(j)\hat{b}_j(O_{t+1}) \tag{A.17}$$

Where $\rho_t(i,j)$ is a nonlinear function of $\hat{\alpha}_t(k)$ and $\hat{a}_{kj}$ for $1 \leq k \leq N$. This new parameter introduces flexibility into the HMMs, similar to a nonstationary HMM in which the transition probabilities depend on time, but now they also depend on the observation sequence. $\rho_t(i,j)$ is defined as $\frac{d_t(i,j)}{\hat{\alpha}_t(i)}$, where $d_t(i,j)$ is the difference between the fuzzy measures corresponding to i and j as calculated in the Choquet integral.

Similarly, the Viterbi algorithm iteration becomes:

$$\hat{\delta}_t(j) = \max_{1\leq i\leq N}(\hat{\delta}_{t-1}(i)\hat{a}_{ij}\rho_t(i,j))\hat{b}_j(O_t) \tag{A.18}$$

$$\hat{\varphi}_t(j) = \operatorname*{argmax}_{1\leq i\leq N}(\hat{\delta}_{t-1}(i)\hat{a}_{ij}\rho_t(i,j)) \tag{A.19}$$

And the Baum-Welch re-estimation algorithm becomes:

$$\hat{\pi}_i = \frac{\hat{\alpha_1}(i)\hat{\beta}_1(i)}{\sum_{j=1}^{N}\hat{\alpha}_1(j)\hat{\beta}_1(j)} \tag{A.20}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\hat{\alpha}_t(i)\rho_t(i,j)\hat{a}_{ij}\hat{\beta}_{t+1}(j)\hat{(b)}_j(O_{t+1})}{\sum_{k=1}^{N}\sum_{t=1}^{T-1}\hat{\alpha}_t(i)\rho_t(i,k)\hat{a}_{ij}\hat{\beta}_{t+1}(k)\hat{(b)}_k(O_{t+1})} \tag{A.21}$$

$$\hat{b}_j(k) = \frac{\sum_{t=1,O_t=k}^{T}\hat{\alpha}_t(j)\hat{\beta}_t(j)}{\sum_{t=1}^{T}\hat{\alpha}_t(j)\hat{\beta}_t(j)} \tag{A.22}$$

This generalized fuzzy HMM approach relaxes the statistical independence assumption required in traditional HMMs, but can be defined to reduce to traditional HMMs. Please refer to [97] for the complete derivation of these algorithms.

Appendix B

Hand Gesture Classifications

The four dichotomies introduced by Nespoulous and Lecours can be used to help classify hand gestures [4]. Each of these scales defines a spectrum of possibilities between two opposing definitions as summarized below:

- Act-Symbol
  Act-symbol gestures can be described on a gradient between pure action, such as lifting a box, and pure symbol such as waving good-bye. The action of lifting a box does not require background knowledge of symbols and meanings or semantics, whereas waving good-bye cannot be understood without a relationship between symbols and meanings (semantics).

- Opacity-Transparency
  Opaque gestures are difficult to interpret without contextual information, including cultural background, or common experiences. At the other end of the scale, transparent gestures are easily interpreted without additional information. Most symbols in the ASL are opaque since it is difficult to interpret these symbols without training or additional information.

- Autonomous Semiotic-Multisemiotic
  Autonomous semiotic gestures are gestures that stand on their own, whereas multisemiotic gestures complement other modalities of communication. Most hand gestures in the ASL are examples of autonomous semiotic gestures.

- Centrifugal-Centripetal
  Centrifugal gestures are focused on or intended for a specific object or person, such as pointing at a box while communicating with someone. Centripetal gestures are not focused on a specific object and not directed or intended for a specific person, such as a shrug when thinking to oneself.

Gesticulation or gestures used in conjunction with speech are commonly broken into the following categories [6, 7]:

- Iconics
  Iconic gestures are transparent gestures that physically describe the object or situation being discussed by the motion or pose of the hands. A concept is often described using its characteristic shape or representative feature. These types of gestures are sometimes referred to as *Mimetics*, including use in [4]. This category is further divided into strictly mimetic gestures and connotative gestures.

  - Strictly Mimetic
    Strictly mimetic gestures describe the object by *drawing* the main outline or representative lines in the object or action in space. Wiggling two fingers and moving the hand forward to represent walking can be considered a strictly mimetic gesture.

  - Connotative
    Connotative gestures describe the object by representing one of the secondary features of the object or action. A frequently used example is describing a goat by its beard.

- Metaphorics
  The motion or pose of the hands in metaphoric gestures suggest the object or situation being discussed. These types of gestures are opaque and are sometimes referred to as *arbitrary* [4].

- Deictic
  Deictic gestures are transparent gestures used to direct the focus of attention to objects. These gestures can be divided into three sub-categories, specific, general, and functional. Specific deictic gestures direct the focus of attention to a specific object, general deictic gestures direct the focus of attention to a class of objects, and functional deictic gestures describe intentions associated with an object.

- Beats
  Beats are unique to gesticulation and are used to emphasize a portion of speech or to *undo* mistakes in speech (Some classify the *undo* gestures separately from beats [6]).

- Cohesives
  Some researchers make a distinction between beats, which are used to emphasize discontinuities in speech and cohesives that are used to emphasize continuity in speech [6].

Appendix C

Hand Tracker

The hand tracker used in this research is a Flock of Birds[1]. This tracker uses a stationary directional pulsed electromagnetic emitter in a known location, and measures position and orientation using a receiver attached to the wrist (see Figure C.1). Although this Flock of Birds provides several alternatives for orientation representation including 3x3 matrix and Euler angles, a quaternion representation avoids loss of precision at certain orientations, or gimball-lock. These orientation-dependent precision variations can occur if Euler representations are used and two axes are equivalent to one another. This results in only two degrees of freedom to represent a three-dimensional orientation, and is possible in Euler representations since rotations are considered independent, yet have to be measured and applied in sequential order.

The gimball-lock scenario can be explained using a set or Euler angles represented with a rotation about the X axis, followed by a rotation about the Y axis, then a rotation about the Z axis. Since the rotations have to be presented in sequential order, the object is first rotated 30 degrees (an arbitrary value) about the X axis. Then, the object is rotated exactly 90 degrees about the Y axis. At this time, the previous rotation about the X axis is now lined up with the Z axis. Any rotation of the object about the Z axis is now along the same axis as the original 30 degree rotation about the X axis. The object can only be rotated about two independent axes in this scenario, as the X and Z axes are equivalent.

Quaternion representations for orientation, although not as intuitive as Euler representations, avoid the loss of precision by defining orientations using the equivalent to an arbitrary axis of rotation and an angle of rotation around that axis [137]. This representation method is an extension of the complex vector space, using i, j, and k, all square roots of -1, in addition to a scalar (w) as shown in Equation C.1. Unit quaternions ($w^2 + x^2 + y^2 + z^2 = 1$) conveniently represent 3D orientations without orientation-dependent precision variations.

$$q = w + xi + yj + zk \tag{C.1}$$

[1] Flock of Birds is a registered trademark of Ascension Technology Corporation

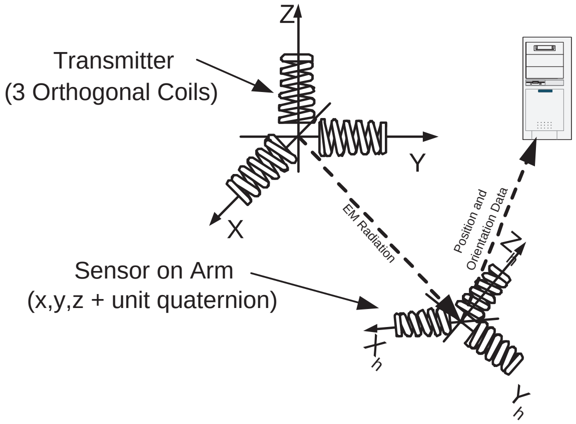
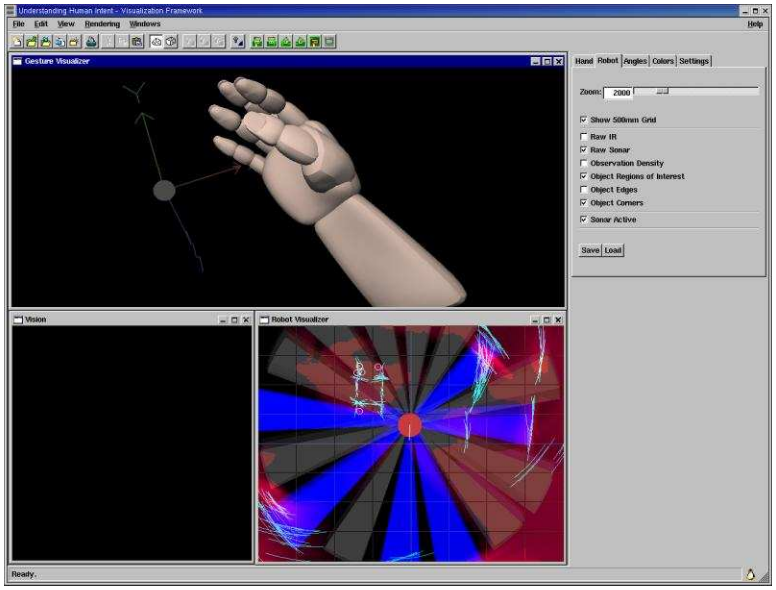


Figure C.1: Hand Tracker

Appendix D

Software

Software developed to support this research includes a hand gesture data acquisition server, a facial expression acquisition component, a robot control and perception framework, the recognition, integration, understanding and conflict resolution algorithms, and a visualization tool shown in Figure D.1. All graphical user interface (GUI) components were built using the FOX toolkit for cross-platform support, and all network and serial communication takes advantage of the GNU CommonC++ abstractions to also provide cross-platform support.



Zoom: 2000

☑ Show 500mm Grid

□ Raw IR

☑ Raw Sonar

□ Observation Density

☑ Object Regions of Interest

□ Object Edges

☑ Object Corners

☑ Sonar Active

Figure D.1: Software has been developed to visualize current and previously recorded hand gesture and robot perception information. This is an invaluable tool to visually inspect, replay, slow down, and qualitatively analyze experimental data. Information about facial expressions is brought together with hand gestures in this software to help understand human expressions.

Nomenclature

Glossary

gesticulation

Hand gestures formed solely to support speech. These gestures are not self-supportive, as they seldom express complete concepts on their own.

gesture

A visible action expressing intent. A motion of the body that contains information. In this thesis, the term gesture may refer to either actions of the hand or the face.

haptic

Based on or relating to the sense of touch (tactile). A haptic feedback device stimulates the skin using physical contact.

modality

A communication channel, or method of expression. Examples of modalities include speech, facial movements, and hand gestures.

mode

A method to interpret human expression. Examples of modes include visual, auditory, and tactile.

multimodal

More than one modality or communication channel. An example of multimodal communication is a spoken word (visual and auditory) together with a pointing hand gesture (visual and possibly tactile).

pervasive

Spread throughout. Pervasive computing devices appear throughout society, including personal digital assistants and mobile phones.

ubiquitous

Transparency. Ubiquitous computing environments are invisible and provide functionality without being intrusive.

List of Acronyms

American sign language (ASL)

A self-sufficient language expressed primarily with dynamic hand gestures and facial expressions.

action unit (AU)

A defined low-level component of the facial action coding system. Some examples of AU's include AU1 = inner eyebrow raised, and AU38 = nostril dilation.

case based planner (CBP)

conceptual dependencies (CD)

conceptual graph (CG)

coupled hidden Markov model (CHMM)

trapeziometacarpal (CMC)

The CMC joint is a joint on the thumb located closest to the wrist.

Common Object Request Broker Architecture (CORBA)

distal interphalangeal (DIP)

The DIP joint is the finger joint closest to the fingertip.

deoxyribonucleic acid (DNA)

dynamic time warping (DTW)

facial action coding system (FACS)

An anatomical based system for measuring visible facial activity. [20]

fuzzy hidden Markov model (FHMM)

fuzzy inference system (FIS)

field-programmable gate array (FPGA)

graphical user interface (GUI)

Hamburg Notation System [99] (HamNoSys)

hidden Markov model (HMM)

infra-red (IR)

metacarpophalangeal (MCP)

The MCP joint is a joint in the hand where the finger meets the body of the hand (the knuckle).

parallel hidden Markov model (PaHMM)

proximal interphalangeal (PIP)

The PIP joint is a finger joint just beyond the knuckle.

past-now-future (PNF)

transmission control protocol (TCP)

time-delay artificial neural network (TDNN)

Taiwanese sign language (TSL)

time varying parameter (TVP)

Index

Colophon

This document has been typeset in LaTeX using the memoir class with a custom chapter and page style based on the ruled style. University of Waterloo thesis guidelines are applied, resulting in a $1\frac{1}{8}$ inch gutter margin and 1 inch top, bottom, and outer margins. Headers and footers are placed inside the margins, but no text is closer than 15mm of an outer edge. Since the maximum line length is used, an 11 point text size with a 5 point leader has been selected as recommended in the University of Waterloo Graduate Thesis Regulations.

Figures were designed using a combination of Till Tantau's TikZ/PGF package for LaTeX and Microsoft Visio 2003. Since Encapsulated PostScript is no longer an export option as of Visio 2003, figures are converted to encapsulated PostScript by first printing to the Adobe generic PostScript printer driver to generate a PostScript file, and then converting this file to a conforming encapsulated PostScript format using George Cameron's ps2epsi tool or Ghostscript. The Adobe generic PostScript printer driver options used include: Optimize for portability, and download TrueType fonts in outline format. It was found that printing in landscape mode, and nonstandard paper sizes result in unacceptable output from Visio 2003. If landscape mode or other paper sizes are used (i.e. A4), Visio often sends the figure as a bitmap of unacceptable quality. It is also important to avoid Visio 2003 before SP1, as the last few pixels of each text box are trimmed, resulting in damaged letters at the end of descriptions.

The TikZ/PGF package provided a programmable interface for describing figures with sufficient flexibility to prevent the tool from constraining the design or layout of the figures. This package was primarily used to lay out graph based figures (i.e. conceptual graphs and hidden Markov models). Bitmap images were converted to PostScript using sam2p and jpeg2ps.

Plots were produced using a combination of gnuplot for directly available data and R when statistical analyses were required.