

Multimodal Human Computer Interaction with Context Dependent Input Modality Suggestion and Dynamic Input Ambiguity Resolution

N. S. Sreekanth^{#1}, N.K Narayanan^{*2}

^{#1}Joint Director, C-DAC Bangalore, #68 Electronics City, Bangalore 560100, Karnataka, India

^{*2}Professor, IIIT- Kottayam, Kerala, India

¹nssreekanth@gmail.com, ²nknarayanan@gmail.com

Abstract - This paper reports a novel approach for enhanced implementation of a practical multimodal interface system with context based input modality suggestion and dynamic input error correction or ambiguity resolution algorithm. The context based input modality suggestion algorithm suggests the user to switch over to alternate modality in adverse environment. The dynamic input error correction module helps the user to correct the omission or resolve the ambiguity in the primarily communicated message by asking for the clarification from the user. If the user provides the input corresponding to reported error, the system completes the operation without asking for a fresh start. Tricolor Finite State Transducers (T-FST) introduced in this paper, analyze the semantics of the communicated multimodal message. The strategy adopted for grammar definition provides a wider operational space for the users to interact with the computer system. A T-FST based message understanding module emphasis on completing the desired operation rather than giving importance for recognizing the each and every signal from the input channel. The proposed architecture is tested with a standard set of operations used for basic human computer interaction.

Keywords - Human Computer Interaction, Speech Recognition, Gesture Recognition, Multimodal Interaction, Man Machine Interaction.

I. INTRODUCTION

When human beings communicate with each other we use various modalities like audio (speech) and visual artifacts (gestures, text, and images in various combinations) as a major form of communication. In addition to this human also perceive the smell, taste and haptic signals from the environment which also plays a major role in decision making and knowledge creation. Human cognitive systems are capable of recognizing, synchronizing and understanding the combination of various input signals from different input modalities (sensory organs). Providing human like capabilities for information processing in machines is a topic of research for past few decades. The significant progress in

the areas of automatic speech recognition, natural language processing and computer vision, facilitate the man-machine interaction process more efficient. Combining these technologies for building the user interfaces by mimicking the human way of communication, lead the researchers to think about developing multimodal interface. Multimodal interaction is a type of Human Computer Interaction (HCI), which combines multiple modalities or different modes of communication like speech, gestures, text and various other combinations. The most common multimodal interface combines visual modality (e.g. display, keyboard, and mouse) with voice modality (speech recognition for input, speech synthesis and recorded audio for output). Multimodal systems are sometimes designed based on one main modality, and the other modalities are simply added on top of it [1]. The logical synchronization of recognized input signals from independent input channels and extracting the semantics of the communicated message is really challenging for multimodal researchers. Providing more input modalities for HCI, not only increases the bandwidth of communication, but also helps in resolving the ambiguities in the primarily communicated message. The ambiguity in one mode of signal can be resolved by the other mode of signal. The best examples are using visual information to understand the ambiguous speech (lip tracking for improving the accuracy of speech recognition) [4] -[7].

Most of the multimodal systems reported in the literature are built around speech based input as a major modality for interaction, and other modes like hand based gestures, pen based gestures, brain computer interface are implemented as an auxiliary or a supporting system. As speech is the major mode of interaction, natural language processing and natural language understanding models play a vital role in understanding the communicated message [3], [8], [9]. The performance of the system will not be acceptable, especially in the noisy environment or situational impairment cases where speech cannot be used as a major mode of interaction [9]. Since the message understanding module of the existing multimodal implementations rely on natural language processing methods and rules, the



performance of the system may not be guaranteed if user switches over to alternate available input methods. The present work is motivated by the little attempt reported in the area of multimodal implementation with context oriented input modality and grammar selection so as to ensure the widest range of input patterns for interacting with computer systems using multiple modalities. Similarly, in an adverse environment, there may be a chance of partial recognition of input stream from certain modalities. It is not sufficient for completing the desired operations. This makes the issued command void and user have to give a fresh start for performing the operation. Instead, the error or ambiguity in the input stream can be corrected dynamically. In this paper, we report a novel method to enhance the existing multimodal interface implementation through two functionalities viz. 1) Multimodal interface implementation with context dependent input modality suggestion 2) Dynamic correction of error or ambiguity in the input stream so as to complete the desired task without nullifying the initially issued commands in the man-machine interaction scenario. These two methods enhances, the popular implementations reported across literature [3], [8], [14]. The proposed method improves the performance of the system in order to accomplish a desired task through a wide range of input patterns issued through multiple input modalities. This feature ensures the reliability of a system for completing a desired task, even if certain elements of communicated messages are not recognized other than the keywords and arguments corresponding to the operation. For multimodal message understanding a Tricolor Finite State Transducers (T-FST) is introduced in this paper. T-FST is a modified version of Finite State Transducer which has all attributes of FST with an additional color code imposed on the state. There are three different types of state defined for message understanding and checking the validity of the communicated multimodal message. In order to validate and understand the communicated multimodal message and perform the operation, the proposed T-FST does not require every individual element of the message which is communicated. The system completes the operations with minimal set of input. The message understanding through T-FST ensures a wider operational space for interaction.

The paper is organized as follows. Section II describes the related work in the area, and the Section III discusses about the proposed architecture of multimodal system with context dependent modality selection algorithms. The dimensionality of a multimodal message and the density of an input channel in a multimodal message are defined in Section IV. How the modality density information is used for environmental depended modality selection is discussed in the Section V. The multimodal message generation without temporal information and with temporal information is discussed in Section VI. Section VII describes the multimodal message understanding algorithms. The strategy adopted for multimodal grammar definition is discussed in the beginning of this section. A tricolor-finite state

transducers is introduced which are used for understanding and translating the multimodal message to system understandable commands. Dynamic correction of error or ambiguity in the input message is also explained in this section. The algorithms discussed in this paper are validated with a use case of basic computer interactions through multimodal interface. The experimental details and results are discussed in Section VIII.

II. RELATED STUDY

The implementation challenges of multimodal interaction system can be broadly categorized into two. The first one is recognizing the signals from individual input channels (speech, gesture, gaze, movement pattern and other modalities), and the second one is to fuse and understand these heterogeneous data types. The recognition aspects of signals from various individual channels are addressed and reliable results are guaranteed under noise free and less-noisy conditions. The work done by Michael Johnston and Srinivas B, reports a multimodal based techniques to improve the recognition accuracy of automatic speech recognition system by using gestures as an augmented modality under various noisy conditions [2], [3]. As mentioned in the section I., most of the reported multimodal implementations rely on speech based modality and other input modalities are treated as auxiliary or supporting channels [2]. Hence the multimodal message understanding is considered as natural language processing problem and the rules of natural language processing methods are implemented for message understanding. In order to take the full advantage of multimodal implementation the system should work and perform well in different environmental situations with the help of all implemented input modalities [10], [11]. If input from one of the modality is missed or not performing as expected the system should switch over to other available channels for accepting input based on the context. Context-sensitive methods are used in multimodal data modeling for emotion classifications, using audio-visual data [12], [13]. For a wide range of operational environment, the multimodal system should be able to work with the available input modalities independently so that absence or poor performance of particular modality will not affect the overall performance of the system. In multimodal implementation the task completion rate is considered as a performance measure rather than the recognition accuracy of individual modality units. One of the popular practical implementation by Michael Johnston, et.al, reports 25 % improvement in the task completion rate compared with the recognition accuracy of individual input channel [2]. In their study, they have implemented a tablet-computer based multimodal environment, MATCH (Multimodal Access To City Help), where speech and gesture based input modalities are implemented for interaction. Under noisy conditions (noisy for acoustic signals) user himself has to choose the appropriate input modality for interacting with the system, i.e gesture. The work done by Potamianos A, et al.,

conducted an exhaustive study on identifying the dominant modality in multimodal interaction patterns during the training phase with respect to a user. This knowledge is being used by the system to adaptively track the most probable interaction mode and to suggest that modality to the user during the post training phase [24]. This work was further extended by ManolisPerakakis and A Potamianos, where a speech and visual input methods (click) are implemented in an e-form filling task. Here the suitable input modality is suggested to use for performing the task, by analyzing user interaction patterns in order to reduce the interaction time to complete a given task [25]. Little effort is reported across the literature which automatically suggests an appropriate input modality to the user for interacting with the system based on the environmental conditions (noise conditions). The method proposed in this paper implement an automated modality selection algorithm that advises users to switch over to reliable modality depends on the context. As discussed earlier the performance of a multimodal system is being measured against the task completion rate, rather than measuring the recognition accuracy of input signals from different input modalities. Hence, in a noisy environment with partially recognized input stream, a method for dynamically correcting the error or ambiguity in the input stream, without nullifying the issued command is also proposed. The method proposed in this paper; extract the semantics of multimodal message from minimal input patterns so that interactions become more natural and informal. This is achieved through a different strategy that we adopted for implementing message understanding module, which differs from the available implementations [8], [9]. The proposed method gives more importance for completing a desired operation rather than giving importance for recognizing each and every unit of communicated messages through various input modalities. In this method system complete the operation with minimal input. A tricolor finite state transducer is introduced in this paper and is used for understanding the communicated message. The proposed method supports the input modality implementation that does not follow the formal rules of natural language processing like, gesture based input. In such cases the system has to operate on the limited input information to complete a desired task.

III. THE ARCHITECTURE OF CONTEXT DEPENDENT MULTIMODAL SYSTEM

The architecture of a multimodal system with a context dependent modality selection proposed in this work is shown in Fig.1. The system comprises of recognizer unit, word lattice generator, multimodal message generation module, environmental analysis and input modality suggestion module, multimodal message understanding module and input/output command generator.

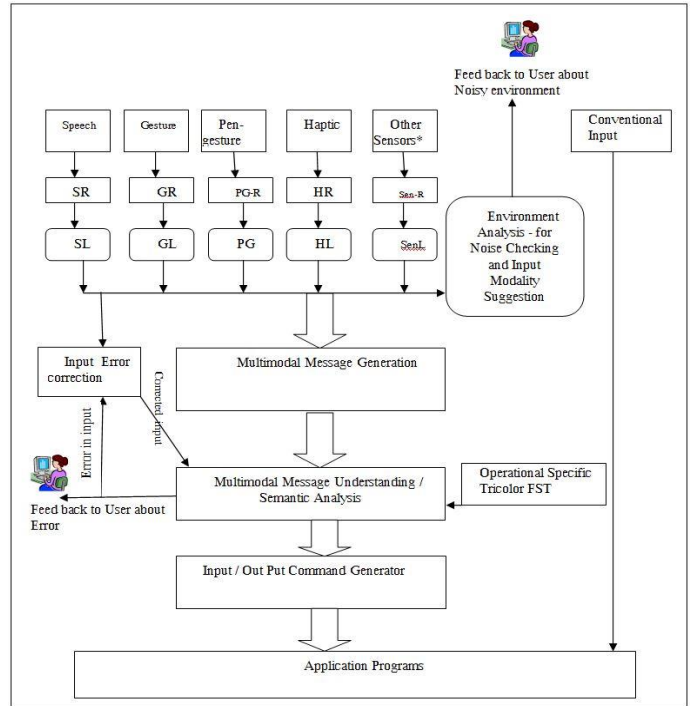


Fig. 1. Architecture of Proposed Multimodal system.

*The Sensors are electro chemical sensors (for Smell , taste). [SR-Speech recognizer, GR- Gesture recognizer, PG-R Pen gesture recognizer, HR- Haptic Recognizer, Sen-R Other electronics Sensor recognizer, SL, GL, PGL, HL, SenL are the lattices corresponds to Speech , gesture, pen gesture haptic and other electronic sensors].

The input signal; i.e., digitized signal from various sensors (camera, microphone, touch screen, chemical sensors and other electronic sensors) will be sent to the corresponding recognizer unit. The recognized signals will be given to the lattice generation module for generating the corresponding word lattice with the appropriate timestamp as discussed by Johnston, et al [3]. The generated word lattices will be given to the multimodal message generation module for generating multimodal message. From the generated word lattices, the system calculates the modality density corresponding to each input channel. The environment analysis module will analyze the noise level corresponding to each channel. If the prominent channel used for communication is found to be noisy, the system reports this to the user and suggest the user to switch over to alternate modality. The environmental dependent modality suggestion module is the key feature of the proposed architecture which differs from the multimodal implementations reported in literature. The environment analysis module will continually monitor the environmental parameters and it provides the appropriate feedback to the user. Once the user starts interacting with the system, the multimodal message will be generated. The logically fused word lattice with temporal information represented using a markup language notation is known as multimodal message. The generated multimodal message will have candidate

elements from various modality sets corresponding to each input channel.

The generated multimodal message will be sent to the multimodal message understanding module for extracting the semantics of communicated messages. This module will generate the parse tree and the semantics of message will be analyzed. Since the basic operations with computers are simulated as part of this experiment, T-FST will generate the equivalent operational command with parameter list as an output corresponding to the communicated multimodal message. If any error or ambiguity in the communication is found, it will be notified to the user for correction. The error or ambiguity in the input also to be notified to the input error correction module, so that this module will listen to the lattice generator for input. If the input is received within the stipulated time, this will be directly given to the message understanding module bypassing the multimodal message generation module. This enables the user to dynamically correct the error or ambiguity in the communicated message. For example, the user issues a command "Delete this file" via speech and also expected to provide the information about which file is to be deleted. The information corresponds to "this" can be provided through hand gesture or pen gesture. If the user did not provide the information, the command seems to be void and this will be reported to the user that "the file to be deleted is not mentioned". The system waits for a specified time so that the user can issue the missing parameter corresponding to the command through any of the input channels. Once input is received, the system completes the operation. If the environment is noisy for speech based input, the proposed system encourages users to use the gesture based method for interaction. For example, when the user issues a command "delete this file" and also provides the pen-based gesture equivalent to delete, and also shows the file to be deleted as a pointing gesture or a pen based gesture, the system performs the operation; if, at least from one of the sensor input stream is recognized and, the operation is valid at that context. The system also provides feedback about the noisy environment for speech based input and suggest the user to switch over to alternate modality. The input-output command generator will generate the corresponding command to interact with the system. The mapping of commands from user vocabulary space to system vocabulary space will be done by the input-output command generator. This will interact with the application programs. The conventional devices (Keyboard and mouse) will interact with application programs directly. The proposed method is tested and validated against the basic human computer interaction operations. The following section discusses certain features of multimodal message i.e, dimensionality of a multimodal message and density of an input modality for a given multimodal message which in turn to be used for context based modality selection.

IV. DIMENSIONALITY OF MULTIMODAL MESSAGE AND MODALITY DENSITY.

The dimensionality of multimodal message is a measure of the number of participating input modality / input channel to form a given multimodal message. This information is used for calculating the modality density corresponding to each input channel. The modality density measures the prominence of each channel over communicated multimodal message. The following section gives details about the dimensionality of multimodal message.

A. Dimensionality of Multimodal Message.

The dimensionality of multimodal message is defined as the number of participating modalities set for composing the multimodal message. Let M be a multimodal message which is defined over N input modalities (input channel), i.e., the multimodal message M composed of elements or symbols from the N different input set.

Let $X_1, X_2, X_3, \dots, X_n$ be the N different modality sets from which candidate element are drawn for composing a multimodal message M. The multimodal message

$M = a_i \oplus a_j \oplus \dots \oplus a_k \#$. The operator \oplus is a string concatenation operator which concatenates string generated by the recognizer unit corresponding to each input channel. The symbols $a_i \in X_i, a_j \in X_j, a_k \in X_k$ and "#" is terminal symbols or end of message for a given stream in a given context. The message will be processed after receiving the # symbol defined as the end of the message. The terminal symbol will be generated by the system automatically after a given time interval to conclude that particular message is complete. The dimensionality of a multimodal message $D(M)$ is defined as

$D(M) = \text{number of participating sets for forming the message.}$

$D(M) = \sum_{i=1}^n d_i$; where $d_i = 1$ if Message M has an element from X_i else $d_i = 0$

A spatio-temporal plot is obtained for understanding the distribution pattern of input signals from N input modalities or channels. The horizontal axis represents the duration for which a particular modality is present and the vertical axis represents the presence of corresponding input modality during that interval.

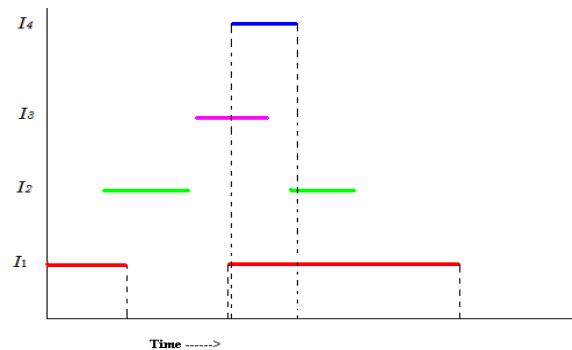


Fig. 2. An example Spatio-temporal Graph

Fig.2 shows a generic spatio-temporal plot for multimodal message where I_1, I_2, I_3, \dots are the input modality-1, input modality-2,..... The given spatio-temporal graph is a Boolean graph which gives the temporal presence/absence of input signal in the given space. Each input modality, i.e., I_1, I_2, \dots, I_n are independent, so that the system expects the signal from these different channels independently. The following section describes how to calculate the modality density using spatio-temporal plot.

B. The Modality Density

The modality density of an input modality in a given multimodal message is defined as the ratio of the duration of the presence of a particular modality over the overall duration of the multimodal message. So the density of k^{th} input modality is

$$\prod_k = \frac{\text{Total Time duration of } I_k \text{ in a given Multimodal Message}}{\text{Total duration of Multimodal Message}}$$

In Fig.2, I_1 is the dominant input modality in the multimodal message followed by I_2 then I_3 and I_4 . It is also possible that the user can provide input through one or more channels simultaneously. For example, when a user says a command "delete this file", the pointing of the file to be deleted can happen during user speaks the sentence, or it can happen little before, or immediately after the speech based input. The following session describes how the modality density information will be used for recommending the appropriate input modality based on the environmental noise conditions.

V. CONTEXT BASED INPUT MODALITY SUGGESTION

The design of context based input modality suggestion module for multimodal implementation is inspired from the human-human communication model. In a noisy environment, for example, inside a factory, when two people communicate each other, the initial mode of communication may be speech based. Depending on the response from the counterpart (if unable to hear) the mode of communication may be switched over to gesture based. The context dependent modality selection algorithm calculates the density values for each input modality in the communicated message. The density of each input modality will be calculated based on the input from the word lattice, from which the dominance of a given modality can be identified. If the noise level for the prominent input modality is beyond a threshold, then the system asks the user to switch over to other suitable modality. If all channels are noisy, system suggests user to use conventional methods for For a given multimodal system N numbers of input modalities are integrated for interaction. At a given context, the message M has elements from all these N input modalities. The algorithm presented in Fig. 3 suggest user to use appropriate input modality for interaction, based on the environmental parameters.

Algorithm –Context Dependented Input Modality Selection
 Let M be the Multimodal Message, N be the number of input channel where system is listening for input. IP_M_i indicate the i th input modality

1. Calculate \prod_i for all input channels of the generated Multimodal Message M
2. Choose the i such that Max $\{\prod_i\}$ (use the IP_M_i)
3. Calculate the noise level μ_i
4. If $\mu_i >$ Threshold
 then
 Provide feedback to user about the noisy environment
 exclude the i -th IP_M_i and find the next densed channel as mentioned in step 2.
5. Provide feedback about the noisy environment and ask user to use the alternate channel
6. If All channels are noisy then suggest user to switch over to Conventional Input method.

Fig. 3. Algorithm for Context Dependent Input Modality Selection

Once the user starts interacting through the suggested input channels, the multimodal message generation module will concatenate the word lattices from different recognizer units with temporal information and the corresponding multimodal message will be generated.

VI. MULTIMODAL MESSAGE GENERATION

Multimodal messages are logically fused word lattices returned by independent input recognizers and interpreted using markup language notation.. Signals from individual sensors are recognized and corresponding word lattices will be generated. The multimodal message generation module will use the generated word lattices to create multimodal messages. In this study a markup language based format is used for composing the multimodal message [16]. A typical multimodal message begins with <BEG_OF_MES, IP_Set = ($I_1, I_2 \dots$) > where the "IP_set" attribute, lists the participating input modalities $I_1, I_2 \dots$ etc. in the given multimodal message. The word lattices generated corresponding to each input modalities are embedded within the input modality tag < I_i > ... </ I_i >. For example the word generated by the speech lattice is embedded within the tag <s> ,</s>. The word tag, <w1>... </w1> ,<w2>... </w2> etc., contains the recognized words from each input channels which will be embedded with the corresponding input modality tag . So the message structure corresponding to a given input modality will be < I_k ><w1>... </w1> ,<w2>... </w2><</ I_k >. The input modality tag may also have a time attribute, which reflects the temporal details of the event's occurrence (input). In the following parts, we'll look at how to create multimodal messages with and without temporal details. [26].

A. Multimodal message Generation without Temporal Information (Non-temporal Multimodal Message)

Speech, hand gestures, and pen gestures based input techniques are incorporated for communicating with the

device in this example of multimodal message generation without temporal details. Consider the case of removing a file from the system; the user has a variety of options because different modalities are included. Assume the user says, "Delete this file," followed by a hand or finger motion pointing to the appropriate icon. The visual feedback of the selection of user choice will be provided. The string generated for processing will have the candidate members from speech, vocabulary as well as from gesture vocabulary set. A speech recognition system recognises the words "delete," "this," and "file" as members of the speech vocabulary collection S. [26],[27].

```
<BEG_OF_MES, IP_Set = (S,G)>
  <S>
  <w1>Delete</w1>
  <w2>this</w2>
  <w3>file</w3>
</S>
  <G>
  <w1>pointer (200,175)</w1>
</G>
<END_OF_MES>
```

Fig 4. Multimodal Message corresponding to "Delete this file"

Similarly the gesture recognition system will return a string with location reference information, for example point (200,175), i.e icon at (200,175) location is referred by user. This location is referred by "this" word in the speech based input. The vocabulary database for speech, hand gesture and pen gesture used for this experiment are discussed in the Section VII. Figure 4 shows the multimodal message that was created in response to the command "delete this file."

The values S and G, which correspond to speech and hand gesture, are stored in the IP set attribute of the above message. The word lattices formed by various input lattices are embedded within the corresponding input modality tag, resulting in the generation of a multimodal message. Once a multimodal message has been created, it will be sent to a semantic analyzer so that the meaning of the message can be determined. If the user's environment prevents them from speaking, they may use a pen gesture like marking over the file icon "X". Figure 5 depicts the multimodal message that corresponds.

```
<BEG_OF_MES, IP_Set = (PG)>
  <PG>
  <w1>Delete</w1>
  <w2>Point(190,300)</w2>
</PG>
<END_OF_MES>
```

Fig 5. Multimodal message corresponds to Delete a file through pen gesture.

B. The multimodal Message Generation with Temporal information (Temporal Multimodal Message)

In certain circumstances, the time of occurrence of an input event may be needed in order to process and comprehend the semantics of the communicated message. A basic illustration can be used to demonstrate this. When a user issues a command to the system through the speech as "Copy this file and paste to that folder" and gestures for the source and destination corresponding to the deictic "this" and "that" in the utterance.

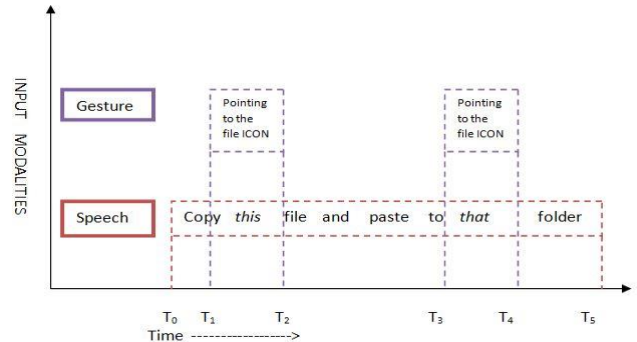


Fig 6 Sequence diagram corresponding to "Copy this file and paste to that folder"

The pointing gesture is used twice in this example, with the first pointing to the source and the second pointing to the destination. The device generates the corresponding multimodal string based on the event's time of occurrence and adds the corresponding time stamp. Figure 6 depicts the sequence diagram for the scenario described above. These two gestures must be ordered chronologically since they state the source and destination directories in that order.

```
<BEG_OF_MES, IP_set=(S,G)>
<S, st="T0", et="T5">
<w1> copy </w1>
<w2> this </w2>
<w3> file </w3>
<w4> and </w4>
<w5> paste</w5>
<w6>to </w6>
<w7> that </w7>
<w8> folder</w8>
</S>
<G >
<w1 st="T1" et="T2"> Pointing (x,y)</w1>
<w2 st="T3" et="T4"> Pointing (p,q)</w2>
</G>
<EOF_MES>
```

Fig 7 Multimodal Message with Temporal Information "Copy this file to that folder"

As a result, the multimodal string should be generated with the event's temporal aspects in mind. For the above tasks, the multimodal message with temporal information can be interpreted as follows. Each event's start and end

times are denoted by the letters "st" and "et," respectively. The "st" and "et" attributes are contained within the input modality tag. Figure 7 shows the generated multimodal message with temporal details. From the above multimodal message the operational keyword, the time stamp and the attributes associated with various tags can be extracted using the transducers defined corresponding to the operation[27].

Multimodal Message Generation Algorithm
 Algorithm – Multimodal Message Generation
 Assumptions: Input channel listeners are implemented threads, M = NULL

1. Accept input from input receptors Independently
2. Recognize the inputs and convert it in to string
 $I_i = (a_i, t_i)$ (Where a_i is the recognized symbol from IP_M_i, ie. ith input modality, t_i is the time stamp)
3. $M = M \oplus I_i$
4. Repeat through step 1 until $\forall I_i = \text{NULL}$
5. *wait(t sec)*
6. If $\forall I_i = \text{NULL}$
then Compose *Multimodal_Message*(Use the notation above mentioned)
else go to step 1 and repeat the process
7. End of Algo

Fig. 8 Multimodal Message Generation Algorithm

We have discussed the method of generating the multimodal message without temporal information and with temporal information along with examples. The multimodal message generation module recognises the heterogeneous signals from different input channels and sends them to the multimodal message generation module to compose the multimodal message. Figure 8 shows the algorithm for implementing multimodal message generation.

VII. MULTIMODAL MESSAGE UNDERSTANDING

Once the multimodal message is generated, it will be given to the multimodal message understanding module for extracting the semantics of the message. We have used a different strategy for multimodal message understanding module implementation which is different from the implementation reported so far. As discussed in the introduction, the conventional multimodal message understanding module implementation relies on the rules of natural language processing. Strong adherence with syntax and grammar are expected in such cases, which may lose the naturalness in communication or it may restrict the informal way of communication. The method proposed in this paper extract the semantics of the message from the minimal input pattern. In other words, system does not require every element in the communicated message in order to understand the meaning. Basic operations for interacting with computers are considered for conducting a case study for the proposed model. In this method more preference is given for completion of a desired operation rather than giving more importance to recognizing the individual signals from independent sensors. This gives a wider flexibility for the user to communicate the message in an informal way. The following section discusses the strategy adopted for

implementing the multimodal message understanding.

A. Strategy for Multimodal Grammar definition and Message Understanding

The motivation for implementing the multimodal interface is not only restricted to provide a rich choice for interaction with systems, but also for providing more naturalness in the human computer interaction. If the syntax of the message used for communication is highly rigid and rule based; the interaction method loses the naturalness (the user may not be able to communicate in the way he/she is used to) and it become robotic. In order to accommodate the various styles of input patterns of different users for performing a given task, the message understanding module should implement a flexible syntax and semantic analyzer. For example, if a user wants to delete a file and command is issued through speech and gesture, the user can say “Delete this file”, and the file icon can be shown via gesture or through pen-gesture. The user can also say “this file delete”, “this one delete” , “this delete”, “delete this ” etc. in all these cases the input corresponding to the deictic *this* is expected to provide as gesture. If the environment is noisy and the use of speech as a major input modality is discouraged by the system, the user can issue the command through a gesture " ✕ " (it can be pen gesture or can be a gesture using fingers) for deleting the file. However, the user can issue speech based commands also in parallel to gesture based input, because the context analyzer continuously monitors the environment so that whenever the noise level becomes less, system responds to speech based commands also. The operation will be initiated by the system if at least one of the input is recognized, i.e., either from speech or from the gesture. In this case the word "delete" is the operational keyword and other words, in the communicated messages are not very important other than at least one argument which is the file to be deleted. This improves the reliability of the system in terms of accomplishment of a given task in adverse environment. The recognition of each and every element in the communicated messages is not mandate, if the objective of the communication is restricted to successful completion of a desired operation at a given context [15]. In this paper, basic desktop/tablet based operations are considered, so that vocabulary and grammar are restricted to the basic system interaction. The grammar is defined based on the operational key words which are used for interacting with the system, like open, close, cut, copy, etc. The function arguments for performing the operations are also to be identified from the communicated message. Tricolor-Finite State Transducers are defined for understanding of the communicated messages through various input channels. The study, reported here implements speech, gesture and pen gesture based communication for interacting with a computer system. When a multimodal message is generated as discussed in the section III., the message will be parsed for deictic resolution. For example, in a generated multimodal message, the deictic reference, i.e., *this, that, it,*

into, etc., will replace with corresponding parameter, i.e., file name which are provided via gesture or pen gesture input. The following section describes how the deictic resolution will be done while parsing the multimodal message[27].

B. Deictic Reference Resolution

For deictic reference resolution markup language based multimodal message will be given to a parser for parsing the message. The parser will generate the deictic reference resolution parse tree. The deictic reference resolution parse tree corresponding to the example “delete this file” discussed in section in Section III is shown in Fig.9. In this case the word after “delete” is “this”, which is a deictic reference whose resolution is done using the three level multi modal reference resolution parse tree. Here the circled word “this” is deictic reference and after applying the reference resolution the reference equivalent to the deictic “this” is “point (200,175)”. In the above example the pen gesture input is not present so it is marked as ∈ (null). A deictic list is prepared corresponding to the language. The string this, that, it, there, here, into, onto etc. are the deictic string for English. The system can look for a corresponding reference location like point (x, y) in the parsed message. The generated output corresponding to the multimodal message after parsing the output string I= "Delete Point (200,175) #", where "#", denote the end-of-the message.

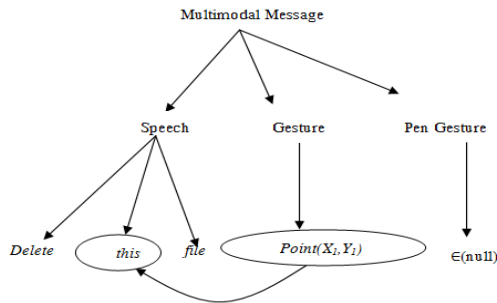


Fig.9. deictic reference resolution Parse Tree for “Delete this file”

Now let us see the processing of a temporal multimodal message, and how the deictic resolution is done using temporal relation. In this case the time of occurrence of each input event has lots of significance for processing the multimodal message.

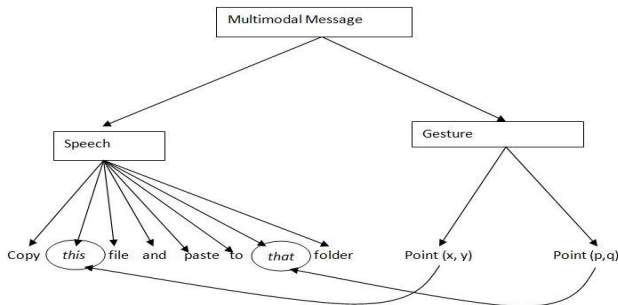


Fig. 10 Deictic reference resolution Parse Tree for "Copy this file and paste to that folder"

For example, if the user issues a command via speech “Copy this file and paste to that folder” and shows gestures for the source and destination corresponding to deictic “this” and “that” in the utterance. Here pointing gesture is used two times. The first pointing corresponding to the deictic “this”, is the source folder which is to be copied and the second pointing gesture corresponding to “that”, is a folder where the file is to be pasted. In order to map the deictic references, this is an argument of the copy function and that is an argument for the operation paste provided through the first and second pointing gesture respectively. In the temporal multimodal message the time of occurrence is also recorded corresponding to input event. The pointing gesture with lower time stamp value will be assigned to this and higher will be assigned to that. The first deictic reference will be assigned to first pointer, then second deictic to second pointer. The reference resolution parse tree for the message is shown in Fig 10. The corresponding output generated I, after reference resolution is I = "copy point (x, y) file and paste to point (p, q) folder # ". After parsing if same operator keywords are found more than once, which are returned by different input modalities, then the system takes only one of them, if it occurs during the same time interval. For example, if a user says "Delete this file" and also shows gesture corresponding to delete operation and also shows the gestures for as location reference to a file corresponding to deictic this, then system discard one of the delete keyword as this occurs during the same time interval and it is treated as duplicate. This message will be given to Tricolor finite state Transducer for message understanding and translating to corresponding operational command. The following section introduces the tricolor finite state transducers which are used for message understanding and it also describes how the message is being understood with a limited set of input[27].

C. Tricolor Finite State Transducer (T-FST)

T-FST is a special variant of finite state transducer (FST) defined for validating and translating the multimodal message to operational commands. Finite State Transducer is a finite state machine which has two tape input tape and output tap. Rather than just traversing (accepting or rejecting) through input string, an FST translates the contents of its input string to output string, i.e., it accepts a string on its input tape and generates another string on its output tape. A FST, F is defined as 6-tuple, $F = \{Q, \Sigma_i, \Sigma_o, i, f, \psi\}$ where Q is the finite set of states, Σ_i is the finite set of input alphabet, Σ_o is the finite set of output alphabets, $i, f \in Q$, which are set of initial and final states respectively. ψ is the transition function for translating the input string to the output string. The transition function is formally defined $\psi \subseteq Q \times (\Sigma_i \cup \{\epsilon\}) \times (\Sigma_o \cup \{\epsilon\}) \times Q$ where ϵ is an empty string. Tricolor-FST, which is a variant of FST, where special colors are assigned to state based on the functionality. In T-FST, the states are classified as three different types, they are Operational Keyword state denote by green Q_g , Parameter state denote by blue Q_b , and Miscellaneous states red Q_r .

The finite set of states Q is defined as $Q = \{Q_g, Q_b, Q_r, i, f\}$. The finite set of input alphabet Σ_i is the vocabulary of the communication, defined as the union of three sets $\Sigma_i = \{T_1 \cup T_2 \cup T_3\}$, Where T_1 is the finite set of operational keywords, T_2 is the finite set of function arguments which includes the entries in the file access table and numbers. The set T_2 will be updated whenever a file or folder is created or deleted during the interaction. The set T_3 is a finite set defined as $T_3 = (T_1 \cup T_2)$, The miscellaneous elements or non-exclusive set of the union of T_1 and T_2 . From any state if it receives an operational key word (copy, paste, delete, etc.), that state will generate the corresponding *opcode* with respect to the operational keyword which are listed in T_1 and will switch over to green state, Q_g . Similarly, if parameters are received, for example, file name, file locations, numbers, etc. which are treated as the function argument for the operational keyword, i.e., *opcode*, then the system will switch over to the blue state Q_b by generating the corresponding function argument listed in T_2 . Whenever a miscellaneous elements are given as input to any state, i.e., the part of communication vocabulary, but neither an element in T_1 nor in T_2 i.e $T_3 = (T_1 \cup T_2)$ system will switch over to the red state Q_r with an output of *empty string*. The attributes of T-FST are defined as follows. The corresponding T-FST is given in Fig.11[27].

- $Q = \{Q_g, Q_b, Q_r, i, f\}$ Finite set of states
- $\Sigma_i =$ set of input symbol $\{T_1 \cup T_2 \cup T_3\}$, The vocabulary of communication
- $\Sigma_o =$ Output symbol {opcodes, function arguments for opcode}
- $i =$ initial states = *idle*
- $f =$ final states = *I/O Prepare (command generation)*
 - Ψ Transition Function defined in Fig. 11. (An element in the transition matrix, $a:b$ corresponding j^{th} row and k^{th} column indicate that, From the current state j upon receiving the string " a " system switch over to the next state corresponding to the column k with an output string " b ". The element # indicate the end of the message). The State Transition diagram is shown in Fig 12.

		OUTPUT STATE				
		Idle(i)	Q_g	Q_b	Q_w	IO-Preparation (f)
INPUT STATE	Idle(i)	$\epsilon : \epsilon$	Op-keyword: opcode	Parameter: argument	Misc: ϵ	Invalid
	Q_g	Invalid	Invalid	Parameter: argument	Misc: ϵ	# : ϵ
	Q_b	Invalid	Op-keyword: opcode	Parameter: argument	Misc: ϵ	# : ϵ
	Q_w	Invalid	Op-keyword: opcode	Parameter: argument	Misc: ϵ	# : ϵ
	IO-Preparation (f)	Invalid	Invalid		Misc: ϵ	Not defined

Fig 11. Transition function

Every state in Q will accept a string and will generate an output string based on the transition function Ψ . How T-FST will be translating the input multimodal message to corresponding operational commands is discussed below.

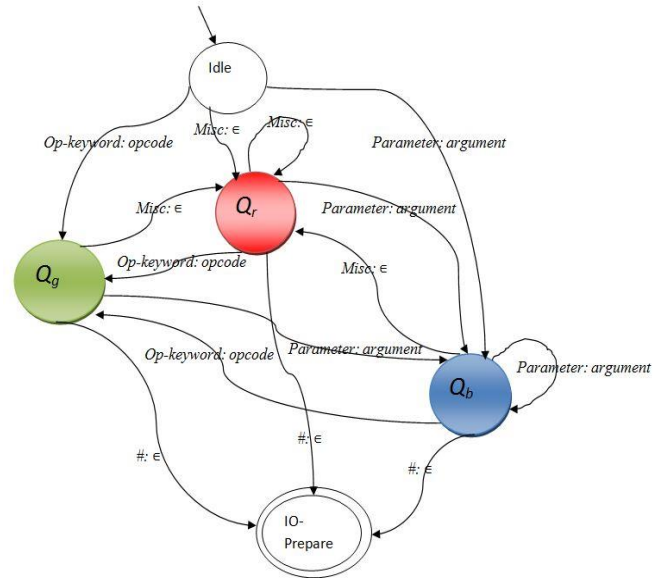


Fig 12. Tricolor -FST for Multimodal Message Understanding

D. Multimodal Message Translation to Operational Commands

The T-FST will receive the parsed message. T-FST will take the input string, which is a parsed string after deictic resolution, denoted as I , and generate the output string Operational commands with arguments. The device will start off in an idle state, waiting for feedback. When the T-FST receives a parsed string I , it accepts it word by word. From idle state (i), upon receiving an operational keyword system will switch over to green state Q_g with an output *opcode* as listed in T_1 . If an argument is received, i.e., the elements mentioned in T_2 , then the system switch over to the blue state Q_b with an output corresponding to the input, i.e., if it is a location reference, the file name at the corresponding location will be generated. This will be the argument for the operator communicated via message. If the input contains an element that is not a member of T_1 or T_2 , it is considered a miscellaneous element, and the device switches to red state Q_r , generating an empty string as an output string. From the state Q_g if system receives an input as parameter, i.e., an element in T_2 , then system generate the output corresponding to the argument as mentioned above. From the colored state, upon receiving "#", the system will switch over to IO-Preparation state (final state). In the IO-Preparation state system generate corresponding system call based on the string generated by T-FST. The label on the arc between any two states denotes the transition function in between the source and destination state. The label on the arc that connect between Q_b and Q_g denotes *Op-keyword: opcode* which

means Q_b accept the input string which is a member of in T1 i.e., *Op-Keyword* and generate the output as corresponding *opcode*. Similarly the transition from Q_b to Q_r denotes *Misc: ϵ* which denotes, the Q_b will accept a miscellaneous string and generate an empty string as output. The state transition follows the transition function mentioned in the Fig.11 and 12.

Let's look at an example of a multi-modal message with speech and gestures as input. In this case, the user speaks the command "can you please open this file" and makes the pointing gesture that corresponds to the deictic "this" in the expression. The parsed message $I = "can\ you\ please\ open\ point(x,y)\ file\ \#"$. The I will be given as input to the T-FST. Initially the system will be in an idle state, when it receives the first element "can" the system will switch over to the red state Q_r , with empty string ϵ as output. The system continues in the same state for next two more tokens "you" and "please", and generate the empty string as output. Currently the system is in Q_r , and when it receives the operational keyword *open*, state transition will happen from Q_r to Q_g with an *opcode fopen* for opening the file or folder. When the system is in Q_g , upon receiving the token *point(x,y)*, the system generate the corresponding file name at the location mentioned in *point(x,y)*, for example "multi.txt" and state transition will happen to Q_b . In the state Q_b , upon receiving the next token *file* system moves to the state Q_r by generating an empty string ϵ . In the state Q_r system accept the string "#", end-of message, the system will switch over to the IO-Preparation state (final) state. The final output generated for the input string $I = "can\ you\ please\ open\ point(x,y)\ file\ \#"$ is "fopen multi.txt". This will be passed on to the system architecture's I/O command generator. The framework also works well when dealing with different arguments. For performing the desired procedure, the multimodal message format will be mapped to a system understandable format (system calls). Operational keywords and its argument list will be extracted from the communicated message using T-FST. The extracted operational key words and its function arguments are mapped to corresponding system calls and the input-output command generator will issue appropriate command. For example the syntax corresponds to delete operation is, DELETE arg1, [arg2, arg3...] (here at least one argument is a must and others are optional) where DELETE is the operational key word. This mapping will be automatically taken care by the transducers defined corresponding to each operation[26],[27].

E. Dynamic Input Error or Ambiguity Resolution

Any error or ambiguity in the communicated messages will be notified to user during the message understanding / semantic analysis. While translating the multimodal message to corresponding system call using the T-FST, the missing argument or missing operational keyword will be notified to user and provides a chance to correct it dynamically without nullifying the issued command. If the

user responds within a stipulated time, the corrected fragment of the input will be directly given to the multimodal message understanding module bypassing the multimodal message generation module so that the desired task can be completed. If the user did not respond within the stipulated time, the incomplete command will be cancelled and the system will enter in to fail state then to an ideal state for accepting fresh input. As discussed earlier, we have implemented speech, hand gesture and pen gesture based input methods for interacting with the system. Initially the system will be in an idle state and whenever an event happens, it will switch to the input acceptance state where three modality states are defined. After certain transition within the state, the input state will switch over to the success state or to the failure state. If it is a failure the feedback will be given to the user about the non-compliance of the communicated message for correction. If the user issues the appropriate input for correcting the ambiguity or error in the primarily communicated message, the system performs the operation and returns to the success state. If input is not received in stipulated time, the system switch over to the failure state, then to an ideal state for accepting new commands. If the semantics of the message cannot be identified, or do not fit in the grammar specified and not at all a valid operation in that context, then the system will switch over to the failure mode. The State transition diagram is shown in Fig.13.

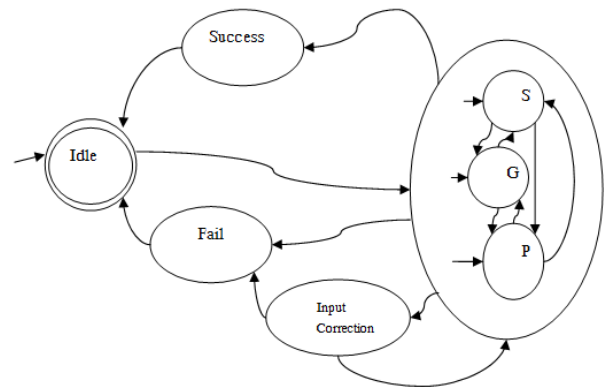


Fig 13. State Transition diagram for a Multimodal System (Input state)

An algorithm for Multimodal message understanding is shown in Fig 14.

<p>Algorithm :Semantic Analyzer for Basic Computer operation Let M be the multimodal input message, Accept the element one after other</p> <ol style="list-style-type: none"> 1. Accept the Next token from M 2. If BEG_OF_MES then Check for participating Input Channels and Initialize the Q_k queues, Q_k is for Input channels for I_k (This will Initialize speech, gesture or pen gesture queues in this implementation) 3. Push the word tag in to the corresponding Q_k queue with temporal Information
--

4. Repeat through step 1 until END_OF_MES encountered.
 5. Prepare bottom up reference resolution parse tree by removing the element from Q queue
 6. If deictic, or reference tag found where element is a member of Q_i , then
 7. Select element from Q_j where $j \neq i$ with appropriate time stamp
 8. Identify the Operator Keyword,
 9. If it is a valid input string(i.e, accepted by finite state transducer) (Based on the format of the operator, i.e, appropriate number of function arguments, This will given in a lookup table.)
 Then go to Step 13.
 10. If any ambiguity in Input pattern
 10.a report to user and ask them to correct
 10.b Listen input lattice generation modules for accepting the missing input pattern
 10.c go to Step 10.
 11. Issue the command string to I/O monitor for execution.
 END_of_Algo

Fig 14. Semantic Analyzer Algorithm for Basic System interaction

VIII. THE EXPERIMENT AND RESULTS.

Multimodal interface for interacting with a computer system for performing the several operations are simulated in this experiment. Basic desktop and tablet operations, which includes file creation, file browsing, file editing operations (text and image files), mail operations, internet surfing, control panel operations, etc. are simulated for evaluating the performance of context dependent modality suggestion and dynamic input error or ambiguity correction algorithm. The speech, gesture and pen gesture based input modalities are integrated for interacting with the system. The performances of the proposed algorithms are tested under different conditions. The details of experiments and results are discussed in section A and B.

A. Experimental Settings.

A list of 50 predefined tasks in abstract level was prepared and given to users for performing the operations with desktop/laptop computers. The task list was prepared based on the operational keywords listed in Table I. Some example tasks are, “copy a file from a folder to another folder”, “open a file and search for a desired word”, “open the browser and search for a desired key word through google”. 30 users (12 female and 18 male) computer professionals within an age group of 25 to 40 were selected for the experiment. The prior information about different input modalities implemented for interacting with the system was provided to users. Users were given a freedom to choose the input method at their convenience for interaction. The performance of the proposed system is evaluated along the three dimensions. They are 1) Environmental dependent modality suggestion, which measures how effectively system recommend user to switch over to alternate modality if the environment is adverse for most frequently used input channel (dominant channel), 2) Dynamic correction of error or ambiguity in the primarily communicated message without nullifying the original message, which measures the

success rate of completion of an operation by seeking clarification from user, if partial portion of communicated multimodal message is recognized 3) Successful rate of completion of desired task through multimodal techniques compared with mere speech based interaction. The listed operations are tested with speech, pen gesture and hand gesture based input methods with different function arguments. The experiment was tested in various signals to noise levels for speech based interface and checked how system suggest alternate effective input modality suitable for interaction based on the current environmental conditions.



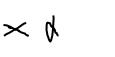

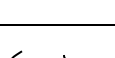

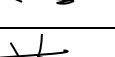

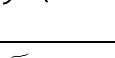



**I. TABLE
 OPERATIONAL WORD LIST FOR THE EXPERIMENT**

Open	OK	Move-down	Print
Close	Cut	Next	Search
Zoom in	Copy	Previous	Find
Zoom out	Paste	Page-up	Mail
Cancel	Move -up	Page-down	Surf
Click	Start	Properties	Select
Save	Save as	delete	redo
sent	attach	change	Select
Go to	Shut-down	Restart	Logoff
Minimize	Maximize	Cascade	Undo

Sphinx4 speech engine is configured and integrated for recognizing the speech based input [23]. For hand gesture recognition computer vision based techniques are implemented. The skeleton extraction method combined with 2D motion vector features are extracted from the video frames for recognizing the hand gestures. Both static and dynamic gestures recognized using this technique [17]-[19]. For pen gesture recognition online strokes are captured and the chain codes are extracted [20]-[22]. User can draw the symbolic representation equivalent to an operation, or user can write operational key word in full or partial in the screen as shown in examples listed in Table II. The forward probabilistic prediction model is used for recognizing the corresponding pen stroke. Hence even user writes the part of the word that also will be recognized.

The noisy environment for speech signals are simulated through adding the AdditiveWhite Gaussian Noise (AWGN) with clean speech at different SNR levels. While interacting through pen/stylus, we have introduced a shaking effect, especially if user interacting with tab in transport vehicle, due to shaking the handwriting may get disturbed, so that algorithm asks user to switch over to speech. If all input channels are disturbed the system encourage the user to use conventional methods.

II. Table
THE EXAMPLE PEN GESTURE DATABASE SNAP SHOT

	Select		Mover up - Page up
	Delete		Movedown/ Page down
	Zoom in		Zoom Select area
	Zoom out		OK
	Copy		Paste
	Cut		Save

The performance of dynamic input ambiguity resolution module is tested in two ways .The first one, the users have been asked to make the random mistake or omissions and found that system asked the user to correct the error. The second one, tested in nosy environment where inputs are not recognized or recognized inputs are not valid at that context. The system reports this and user can provide the right input relevant to that context. In both the cases the omission or unrecognized elements are either operational keywords or the corresponding function arguments. The system also tested for the overall success rate for completion of desired task.

B. Experimental Results

The performance of various input modalities are tested independently for carrying out a desired operation under different noise conditions. Similarly combinations of various modalities are also tested for performing a desired operation. The first set of experiments are conducted for evaluating the success rate of task completion with mere speech based input. In this case dynamic error correction module and input modality suggestion modules are disabled so that the system does not ask for correcting the missing part or unrecognized part of the communicated message dynamically and it also does not suggest to use alternate modality while testing with speech based input. In such context, previously issued commands will be nullified and the user is expected to make a fresh start. The accuracy of the speech based interaction is tested with pronounceable function arguments and unpronounceable function arguments. The pronounceable function arguments are the file names and folder names such as "home", "computer", "Documents" etc. Such cases user can issue operational commands like "open computer" or "go to home folder" etc., to perform desired operations. The unpronounceable function arguments include the abbreviations and short forms which are used for naming the file or folders also add digits or special characters along with the pronounceable arguments. For example "CMU", "MIT", "abc" etc. which are read as "C M U", "M I T", "A B C".

Similarly we also name the folder or file names as "jamesI" and read as "james one". The success rate for operational completion with mere speech based input (Clean speech) with pronounceable arguments is found to be 93% under lab conditions. The word error rate for speech recognition system is 6.4 % (vocabulary size 780 words) The success rate of operational completion with un-pronounceable arguments with mere speech based input is found to be less than 10 % and get failed every time under noise conditions i.e, signal to noise ratio(SNR) less than 50dB . But the recognition accuracy for operational key word alone is 97% for clean speech and it varies from 85 % to 59% with different SNR values varies from 50dB to 10dB. In such scenarios the usage of gesture modality helps user to point appropriate object which is named as per the interest of user like "jamesI", "CMU", etc.,. User can point or mark the unpronounceable object and the operation to be performed can be issued via speech. Even pen gesture can also be used to perform the operation. For example user can say "open"

andwrite *abc* on the screen, then system opens the file "abc". Averages of 94 percentage of time system respond as per the expectation if commands are issued via combination of clean speech and gesture. Under different acoustic noisy conditions i.e, SNR from 50dB to 10 dB, the success rate for task completion varies from 93% to 85% with different combination of input modalities. The success rate of task completion is increased to 98 % when the dynamic error correction module is enabled by assuming a condition that, any one of the implemented input channel is free from environmental adverse conditions.

The performance of input modality suggestion module, i.e., recommendation of appropriate input modality based on the environmental conditions was tested by enabling the dynamic error correction module. Switching over from speech based interaction to hand gesture or pen gesture based interaction and vice-versa were tested in different environmental conditions. The suggestion for changing the dominant input modality for interaction happens only if the system repeatedly fails (three consecutive failures) to understand a given communicated message. This assumes that the dominant input channel used in the communication is noisy or the channel used for communication may not be suitable for that user for interaction. Suggestion for switch over from speech based input to gesture based or to conventional input method was simulated via, adding Additive White Gaussian Noise with clean speech signal at different SNR levels. Since dynamic error correction module was enabled during the testing phase system seeks the clarification for the missing or unrecognized crucial part of the message, inorder to complete the operation. When user uses pen-gesture/hand gesture based interaction shaking effect was introduced as a noise then system suggest user to switch over to conventional input methods.If system repeatedly fails to recognize input provided through a given modality, the context based modality selection algorithm

suggest user to switch over to alternate modality. All the participants prefer the speech dominant interaction. They use gestures for pointing an icon or providing clarification for a missing input or unrecognized input which is asked via dynamic error correction module. The rate of switch over from speech based interface to gesture based method varies from 14 % to 38% for SNR values of 50dB to 10dB. Because of the shaking effect introduced during pen gesture based interaction, 28 % of trial the system suggested switch over from pen gesture based input to speech based input. When user interact with system through multimodal interface with dynamic error correction module , with an SNR of 1 dB for speech signal or introduce an adverse condition for gesture based input, system reports 87% of success rate for completing the desired task.

The functionality of dynamic, input ambiguity resolution module was tested and a success rate for task completion with dynamic error correction module is 98 % under laboratory conditions. In outside lab environment because of poor recognition accuracy of individual modalities, the overall success rate of completing a desired operation is around 87%. The above results justifies the importance of context based modality selection and dynamic input ambiguity/error resolution algorithms in true multimodal implementation.

IX. CONCLUSIONS AND FUTURE WORK

This paper reports several enhancements over conventional multimodal implementations. The implemented methods and algorithms take the maximum advantage of rich choices of input modalities for interacting with the system. All input modalities integrated as part of the system may not perform always as expected especially in noisy conditions. The context dependent modality suggestion algorithm suggest user to switch over to appropriate modality based on the environmental conditions. Speech and gesture based technologies are implemented for interaction and the performance of the context dependent module is evaluated under different environmental conditions. The dynamic input error or ambiguity resolution method implemented as part of message understanding module (semantic analyzer), will help user to correct the omission or ambiguity in the primarily communicated message by asking the clarification from the user. This makes the previously issued command valid for stipulated time and if user provides the input corresponding to the reported error, system complete the operation without asking for a fresh start. The strategy adopted for multimodal grammar definition and message understanding gives more importance for operational completion which differs from the previously reported implementations, which gives more importance for recognizing the whole unit of input signals. This strategy also ensures the interaction more reliable and natural and also provides wider operational space. In the current study we have implemented a multimodal system to perform the standard set of operations on desktop and laptop

computers. This need be extended for a generic environment for interacting with any electronics gadget. The development of the operational specific grammar models for generic environment also can be addressed.

REFERENCES

- [1] Markku Turunen, Jaakko Hakulinen, Anssi Kainulainen, Aleksi Meltola, Topi Hurtig. Design of a Rich Multimodal Interface for Mobile Spoken Route Guidance, Proceedings of Interspeech 2007 - Eurospeech: (2007) 2193-2196.
- [2] Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent Patrick Ehlen, Marilyn Walker, Steve Whittaker, Preetam Maloor, MATCH: An Architecture for Multimodal Dialogue Systems , Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, (2002) 376-383.
- [3] Johnston, M. and S. Bangalore. 2005. Finite-state Multimodal Integration and Understanding. Journal of Natural Language Engineering 11(2) 159-187, Cambridge University Press.
- [4] Dupont, S.; TCTS Lab., Mons Polytech. Inst., Belgium ; Luetin, J. Audio-visual speech modeling for continuous speech recognition, Multimedia, IEEE Transactions on 2(3) (2000).
- [5] Gutierrez-Osuna, R. Coll. Station, Texas A&M Univ., College Station, TX, USA Kakumanu, P.K.; Esposito, Garcia, O.N. Bojorquez, A.; Castillo, J.L. Rudomin, I. Speech-driven facial animation with realistic dynamics, IEEE Transactions on Multimedia, 7(1)(2005) 33-42.
- [6] Tsuhan Chen; AT&T Bell Labs., Holmdel, NJ, USA ; Rao, R.R., Audio-visual integration in multimodal communication, Proceedings of the IEEE, 86(5)(1998).
- [7] Nguyen, D.; Dept. of Electr. & Comput. Eng., Univ. of Toronto, Ont. ; Halupka, D.; Aarabi, P.; Sheikholeslami, A. Real-time face detection and lip feature extraction using field-programmable gate arrays, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 36(4)(2006) 902-912.
- [8] Arianna D' Ulizia, Fernando Ferri, Patrizia Grifoni A Learning algorithm for multimodal Grammar Interface, IEEE Transactions on System Man, Machine and Cybernetics, 41(6)(2011) 1495-1510.
- [9] Arianna D' Ulizia, Fernando Ferri, Patrizia Grifoni, Generating Multimodal grammars for Multimodal Dialogue Processing, IEEE Transactions on Systems , Man and Cybernetics- Part A, Systems and Human, 40(6)(2010).
- [10] Nicu Sebe, Multimodal interfaces: Challenges and perspectives, Journal of Ambient Intelligence and Smart Environments 1 (2009) 19-26
- [11] S.L. Oviatt, Mutual disambiguation of recognition errors in a multimodal architecture, ACM CHI, 1999.
- [12] Angeliki Metallinou, Martin Wollmer, Athanasios Katsamanis, Florian Eyben, Bjorn Schuller, Shrikanth Narayanan, Context-Sensitive Learning for Enhanced Audiovisual Emotion Classification, IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, 3(2)(2012) 184-198.
- [13] Loic Kessous · Ginevra Castellano · George Caridakis, Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis, Journal of Multimodal User Interfaces (2010) Springer 3: 33-48.
- [14] Bruno Dumas, Denis Lalanne, Sharon Oviatt, Multimodal Interfaces: A Survey of Principles, Models and Frameworks, Human Machine Interaction: Research Results of the MMI Program, LNCS 5440 (2009) 3-26.
- [15] Leishman, F, Monfort, V, Horn, O, Bourhis, G, Driving Assistance by Deictic Control for a Smart Wheelchair: The Assessment Issue, IEEE Transactions on Human-Machine Systems, 44(1) (2014) 66 - 77
- [16] W3C multimodal interaction framework - <http://www.w3.org/TR/mmi-framework/>
- [17] Bogdan Ionescu , Didier Coquin , Patrick Lambert , Vasile Buzoiu , Dynamic Hand Gesture Recognition Using the Skeleton of the Hand , EURASIP Journal on Applied Signal Processing 2005: 13, 2101-2109
- [18] Ming-Hsuan Yang ; Ahuja, N. ; Tabb, M. Extraction of 2D motion trajectories and its application to hand gesture recognition, Pattern

- Analysis and Machine Intelligence, IEEE Transactions on 24(8)(2002) 1061-1074.
- [19] Ohn-Bar, E, Trivedi, M.M., Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations, Intelligent Transportation Systems, IEEE Transactions on 15(6)(2014) 2368-2377.
- [20] Hung Yuen, A chain coding approach for real-time recognition of on-line handwritten characters, IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 6(1996).
- [21] Plamondon, R.; Srihari, S.N. Online and off-line handwriting recognition: a comprehensive survey, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22(1)(2002) 63 - 84.
- [22] Tappert, C.C. ; Suen, C.Y. ; Wakahara, T., The state of the art in online handwriting recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 12(8)(2002) 787 – 808.
- [23] (2021) CMU Sphnix : Building Applications with Sphnix 4 - Website <http://cmusphinx.sourceforge.net/wiki/tutorialspphinx4>
- [24] Potamianos A, Ammicht,E, Fosler-Lussier, E., Modality tracking in the. Multimodal Bell labs communicator, IEEE Workshop on Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003
- [25] ManolisPerakakis,Potamianos A., A Study in Efficiency and Modality Usage in Multimodal Form Filling Systems, IEEE Transactions On Audio, Speech, And Language Processing, 16(6)(2008) 1194-1206.
- [26] N.S Sreekanth,et.al., Multimodal Interface for Effective Man Machine Interaction, Media Convergence Handbook; Artur Lugmayr and Cinzia Dal Zotto; Springer, Berlin, Heidelberg, Germany, 2016; 3(2016) 261-281.
- [27] N.S Sreekanth, Enhanced Malayalam Speech Recognition Using Multimodal Techniques, Doctoral Thesis, Kannur University, Kerala, Mar 2017.