7-27-2010

Integrated Framework for Interaction and Annotation of Multimodal Data

Afroza Ahmed

To the University Council:

The Thesis Committee for Afroza Ahmed certifies that this is the final approved version of the following electronic thesis: “Integrated Framework for Interaction and Annotation of Multimodal Data.”

Mohammed Yeasin, Ph.D.
Major Professor

We have read this thesis and recommend its acceptance:

Xiangen Hu, Ph.D.

Aaron L Robinson, Ph.D.

Accepted for the Graduate Council:

Karen D. Weddle-West, Ph.D.
Vice Provost for Graduate Programs

INTEGRATED FRAMEWORK FOR INTERACTION AND ANNOTATION OF MULTIMODAL DATA

by

Afroza Ahmed

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Electrical and Computer Engineering

The University of Memphis

August 2010

ABSTRACT

Ahmed, Afroza. MS. The University of Memphis. August 2010. Integrated Framework for Interaction and Annotation of Multimodal Data. Major Professor: Mohammed Yeasin, Ph.D.

This thesis aims to develop an integrated framework and intuitive user-interface to interact, annotate, and analyze multimodal data (i.e., video, image, audio, and text data). The proposed framework has three layers: (i) interaction, (ii) annotation, and (iii) analysis or modeling. These three layers are seamlessly wrapped together using a user-friendly interface designed based on proven principles from the industry practices. The key objective is to facilitate the interaction with multimodal data at various levels of granularities. In particular, the proposed framework allows interaction with the multimodal data in three levels: (i) raw level, (ii) feature level, and (iii) semantic level. The main function of the proposed framework is to provide an efficient way to annotate the raw multimodal data to create proper ground truth metadata. The annotated data is used for visual analysis, co-analysis, and modeling of underlying concepts, such as dialog acts, continuous gestures, and spontaneous emotions. The key challenge is to integrate codes (computer programs) written using different programming languages and platforms, displaying the results, and multimodal data in one platform. This fully integrated tool achieved the stated goals and objective and is a valuable addition to the list of very few existing tools that are useful for interaction, annotation, and analysis of multimodal data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Labeling multimodal data to machine readable form becomes a necessary perspective in multimodal information processing. As a part of semi-supervised learning, this labeling process is generally known as data annotation task. It is a more time consuming task to annotate text, image, video, or their combinations. This contributes to the classification performance. A number of annotation tools are presently available to speed up the semi-supervised labeling process. Very few of those are considered an integrated version, and none of those have all of the necessary components in same framework. In addition, only a few of those were evaluated through the usability testing to have performance evaluations of the annotation framework. In this study, we analyze user's mental load along with the usability testing.

1.1 Annotation of multimodal data

From the user interaction point of view, data annotation is a means of marking up the multimodal data in order to facilitate the interpretation and the understanding of its content.

According to M. Bordegoni [1] modality refers to a particular way or mechanism of encoding information for presentation to humans or machines in physically realized form. Examples include spoken natural language and its transcription, music, and audio icon. Multi-modality refers to the use of multiple modalities to encode information. For example, the audio medium is often used to convey multimodal information in the form of speech accompanied by background music.

The process of data annotation is not new. Psychologists use to annotate non verbal multimodal data as a preprocessing cognitive experiment design. Based on the modalities two types of annotation scheme can be considered: single mode (Figure 1) and multimode (Figure 2) annotation. In single mode, we can consider text, speech, and image/video to be in separate modes. In multimodal annotation, text, speech, and

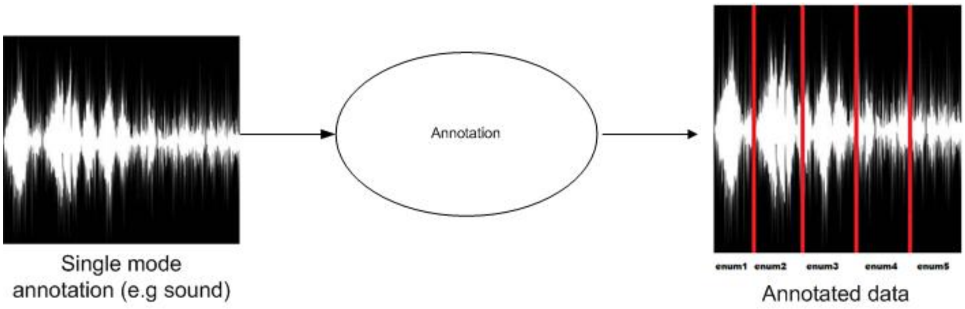


Figure 1: Single-mode annotation

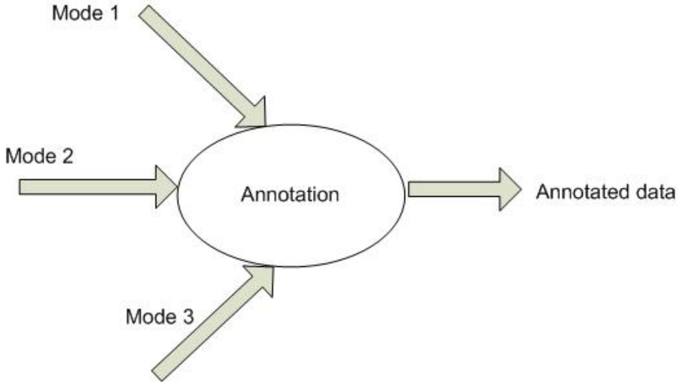


Figure 2: Multi-mode annotation

image/video are considered in the same framework. For example, if researcher has one video data with audio and speech transcription, in multimodal annotation (Figure 3), he can do the annotation of video data along with audio data, and its speech transcription in one environment. Considering the annotation techniques used for annotation, we can consider three types of annotation schemes: expert, semi-supervised, and automated annotation. Annotation only by human experts is the oldest annotation techniques. As human annotators always involve in the annotation task, it deserves high accuracy but less efficiency. Moreover, the human expert should have the proper domain knowledge before start annotating. Semi-supervised annotation technique is relatively easier than expert annotation; the user has to have the knowledge to use annotation tool to annotate the multimodal data. In this case, the annotator can use different software applications. This is a relatively modern annotation technique. A number of annotation tools are available for text annotation, speech annotation, and image/video annotation for semi-supervised annotation. We consider our approach as a semi-supervised annotation scheme that is integrated, but it is not yet completely integrated for every modality. In supervised annotation, the machine should be able to annotate automatically. Very few automatic annotation tools are found for only single mode annotation. Still, they lack multimodal annotation. In Table 1, the three annotation schemes are summarized.

Table 1: Annotation types according to annotation techniques

Annotation Type	**Features**	**Difficulty Level**	**Usage**
Expert	High accuracy, less efficiency, involves human expert	Not easy	To annotate emotion, complex psychological behavior
Semi-supervised	Moderate accuracy, moderate efficiency, involves non-expert human	Moderately Easy	To annotate action e.g., running, walking, and so on, and gesture events
Automated	High accuracy, high efficiency, involves only machine	Not easy	To annotate human motions

Multimodal data annotation is inherently challenging, and some techniques invented so far have limited integration capability considering single mode and multimode. This integrated annotation framework is good enough to annotate and analyze multimodal data. The main function of the proposed framework is to provide an efficient way to annotate multimodal data to create proper ground truth data.

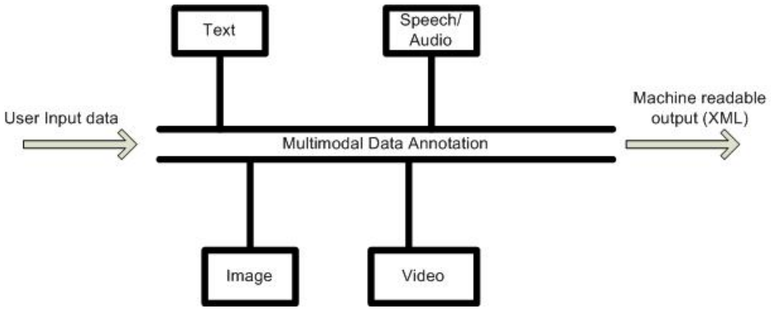


Figure 3: Multimodal annotation

The necessity of a tool to aid multimodal data annotation and analysis has recently increased due to the extensive research with multimodal data analysis. The type of multimodal analysis include all possible verbal and non verbal data analysis, co-analysis and modeling of underlying concepts, such as dialog acts classification, continuous gesture recognition, and spontaneous emotions recognition. This integrated framework is developed with an intuitive user-interface to interact, annotate, and analyze multimodal data (i.e., combination of video, image, audio, and text data) in a single platform. The key motivation for developing the framework is to analyze the data interactively according to predefined requirements such as visual representation of the data, and selecting features from the data. The main challenge is to integrate all these different modalities (text, audio, and image/video) into one environment. It is a challenge to provide this facility to the users for all possible modalities [2].

1.2 Integrated annotation framework

The proposed framework has three layers: (i) interaction, (ii) annotation, and (iii) analysis or modeling.

1.2.1 Interaction

The objective of this framework is to facilitate interaction with multimodal data at various levels of granularities. In particular, the proposed framework allows interaction with the multimodal data in three levels: (i) raw level, (ii) feature level, and (iii) semantic level. As an example, raw level interaction indicates video and audio play back without any processing. In feature level interaction, the features are extracted and researchers can do further modeling and analysis. The semantic level helps to understand the semantics of the data.

1.2.2 Annotation

We are developing a semi-automated annotation system for labeling specified events visible in the video.The events are labeled as specified by the system's requirements. Thus, the system will enhance the productivity of human video annotators and/or cue a subsequent event classification module by marking specified events. The main focus is on the formation of ground truth data, which will provide the basis for answers to many imperative research questions. One of the key motivations for annotation is to provide a convenient way for the researcher to get annotated data from video. The annotation module will save the associated information along with the frame number and event information of the required portion. Manually produced annotations provide the standard against which the performance of automated systems is measured and evaluated. Therefore, the availability of large volumes of manually annotated corpora is a prerequisite to progress in this field.

1.2.3 Analysis

In the context of non-verbal communications, human gestures are dependent on other modalities such as speech or facial emotion. For example, when a person speaks, his hands produce gesture according to the action, emotion, and intention of the speaker. This relation of dependency can be a very important feature of non-verbal communications. In this research, the dependency is modeled through the co-analysis of the modalities. The framework has the module to perform the co-analysis efficiently. It provides a way to look at the signal and sense of the non-verbal communications. Among other non-verbal communications, head nod, body posture, and hand gesture are significant. This research considers the body component tracking and annotation, which helps to perform co-analysis in a better way.

1.3 Challenges

There are a number of annotation frameworks found as open-source freely available applications. Many of those are good for single mode semi-supervised annotation only. Researchers can use those tools to annotate single mode data e.g., text, audio, or video. To combine those separate annotated metadata in order to achieve multi-modality it becomes an extra overhead to the researchers during the cognitive task design. The major problem is binding the separate application which is written in different programming languages. This phenomena raises the question, ”how to reduce the time and binding complexities of multimodal annotation?” This research simply answers this question.

1.3.1 Integration

The framework is developed for Windows platform. Reasons for this choice of platform included built-in support for modern codecs; built-in video and image processing functions from OpenCV[3]; object oriented facilities through .Net framework; availability of high-level frameworks and application programming interface (API); developer tools; ability to combine modern languages (C++, C#, Python, Java, Praat, etc.) in the same application; and to efficiently display all the raw data and featured data in an understandable way. Integrating multimodal data is a challenge because of varying timescales and perceptual characteristics. Dealing with multimedia requires a well thought and robust architecture, including time models, media managers, and players. An additional consideration is the use of video codecs.

1.3.2 Heterogenous Data

In this work, we have data that has a high sampling rate (such as audio/video). Observational data can be either points (i.e., instants in time); these are also discrete data and are made with the limit of precision allowed by digital media. As multimodal data source, we use time synchronized audio and video recordings collected from different sources. Table 2 shows the key modalities we are using by the framework. After completing annotation, we process the data to extract low-level features such as video features derived by computer vision algorithms and audio features such as pitch tracking and intensity tracking. Examples of further processing include speech transcription, hand-tracking, head-tracking, etc. As a result of this process, we have source data with additional feature information.

Table 2: Input data types

Data	Description
Video	raw and compressed collected from various sources
Image	extracted or grabbed from video and /or collected from various sources
Audio	extracted from video and /or collected from various sources
Text	collected from speech transcription and /or collected from various sources

1.4 Motivation

The proposed framework has three layers: (i) interaction, (ii) annotation, and (iii) analysis or modeling. These three layers are seamlessly wrapped together using an intuitive user interface designed based on proven principles from the industry practices. The key challenge is to integrate computer programs written using different programming languages and platforms to display the results, and multimodal data in one platform.

Since this research focused on multimodal analysis, we desired a tool that would allow us to integrate several sources of information. Specifically, we sought to create an integrated system that would give the researcher concurrent access to source data (audio/video), low-level features such as motion points from a video (obtained from automated and semi-automated algorithms), speech meta data, and ground data (through manual annotation). We developed a new software that would support these needs, resulting in a new way to interact, visualize, and analyze multimodal data.

1.5 Outline

The rest of the thesis is organized as follows: Chapter 2 explores the available literature for annotation and analysis framework to provide the research context. Chapter 3 gives an overview of the proposed framework. Chapter 4 and 5 explains the granularity of the annotation and analysis modules. Chapter 6 describes the GUI and explains how the user interface design principles are preserved in the interface and presents the performance evaluation given by a reasonable number of diversified users through collecting their initial reviews with the first phase of the tool. Chapter 7 concludes the thesis by suggesting further improvements to this framework.

2 Background and related works

In recent decades, many annotation and analysis tools have been developed. Some of those are only for one modality such as for video or audio, and some of those are for multimodal data. The following sections give a literature review for these published tools for data annotation and analysis. These tools may have some common features but distinguished themselves with their distinct features. This section also intends to look into those distinctions as well as the common features of these tools. Finally, there is a comparison between these tools and this interaction and annotation framework. From the literature review, the following missing features of these tools are summarized:

- Existing tools did not provide complete coverage of desired features
- Lack of support for compressed video (e.g., MPEG-4)
- Not enough options for heterogeneous data types
- Not enough options for visualization (e.g., displaying data plot)
- Lack of support for co-analysis between video and audio and/or between uni-modal features

Besides, many of these tools are being in continuous modifications or evolutions and some of these have been transitioned to other packages. This reflects the need for a well-developed tool in this data annotation and analysis area. This continuous demands in this particular research area served as the guiding principle for creating this framework.

2.1 Annotation tool

IBM's VideoAnnEx[4] annotates MPEG video sequences with MPEG-7 Meta data framework. VideoAnnEx takes an MPEG video file as an input. The video sequence input is segmented into smaller units called video shots. Each shot in the video sequence can be annotated with static scene descriptions, key object descriptions, event descriptions, and other lexicon sets. The annotated descriptions are associated with each video shot and are put out and stored as MPEG-7 descriptions in an XML file. VideoAnnEx can also open MPEG-7 files in order to display the annotations for the corresponding video sequence. VideoAnnEx is different from this proposed framework, as here any media file can be played and annotated and new modules can be added. The proposed framework includes a video annotation system using subject based comment to annotate key video events.

Anvil (Video Annotation Research Tool)[5] is an annotation tool written in Java that has been available since 2000 and was designed to work with audio and video and provide visualization of supporting metadata. It features frame-accurate annotation and is hierarchical with multiple user-defined layers. It uses color-coding on multiple tiers to represent events, and can annotate links between tracks if desired. ANVIL was created for gesture research and has been applied in other domains (human computer interaction, linguistics, computer animation, etc.). It imports time-aligned speech markup from Praat (sound analysis tool)[6] and XWaves. The major supported video formats are Audio Video Interleave (AVI) and QuickTime. The software is downloadable as a Java executable from DFKI (German Research Center for Artificial Intelligence). Anvil [7] was originally written and is currently maintained by Michael Kipp.

Microsoft's MRAS [8] system is designed to support annotation of multimedia content about a lecture asynchronously. A user can download a lecture along with the comments added by other students and professors. Users add their own annotations and save them onto the annotation server. The MRAS system focuses on users' asynchronous on-demand training.

The Classroom 2000 project implemented a software infrastructure that captures much of the rich interaction during a typical university lecture including all aspects of a lecture in classroom–audio, video, and blackboards. All activities are captured and recorded with timestamps, and students can access the 'lecture' by replaying the recorded video, audio, and slides.

ELAN (EUDICO Linguistic Annotator) [9] is an open source, cross-platform Java tool that was created for psycholinguistics research. It allows one to create, edit, visualize, and search annotations for video and audio data. It includes features such as display of audio and video with annotations, time linking of annotations to media streams, linking of annotations to others, and an unlimited number of annotation tiers as defined by the users, as well as import, export, and search options. ELAN is under active development and is functional; it was made to be extendible and support collaborative annotation/analysis. Many of ELAN's features make it a good example of what can be done using Java on modern computer platforms. ELAN is developed and maintained at the Max Planck Institute for Psycholinguistics.

iVas [10] system can associate archived digital video clips (DVD, TV) with various text annotations and impression annotations using the client server architecture, the system analyzes video content to acquire cut/shot information and color histograms. Then it automatically generates a web document that allows the users to edit the annotations. It is also a standalone system.

2.2 Analysis tool

Begun in 1994, CAVA (Computer Assisted Video analysis) [11] was created for use on both PC and Macintosh: in particular, there were two transcription tools: the Transcription Editor (TED) for use with the PC for transcribing analog video tape, and Media Tagger for working with digital video on the Macintosh. CAVA is a multi-platform system that can access data stored in an Oracle database on a Unix server. In addition, the CAVA tools are platform-dependent, use a proprietary data storage format, and are designed for single-site use (i.e., site-specific). CAVA was created at the Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands.

2.3 Speech analysis tool

Praat [6] is an open source, cross-platform tool for doing phonetic analysis of speech on the computer. It originates from Paul Boersma and David Weenink at the Institute of Phonetic Sciences, University of Amsterdam. Praat has a variety of built-in functions and machine learning algorithms for working with speech (audio), such as spectral analysis (spectrograms), pitch analysis, formant analysis, intensity analysis, etc. Praat provides its own scripting language, permitting additional functions to be added, and can be used to create speech transcriptions. It is implemented in C++ and uses X-windows/Motif, Carbon (for Mac OS), and QuickTime.

WaveSurfer is an open source tool for sound visualization and analysis. It can be used in its default configuration as a stand-alone tool for transcription, or it can be extended through plugins. WaveSurfer can also be embedded in other applications. It uses a toolkit called ”Snack Sound Toolkit”. Both WaveSurfer and Snack are from the Department of Speech, Music and Hearing at the School of Computer Science and Communication, Royal Institute of Technology in Sweden.

2.4 Comparison of selected tools

Our tool is similar to the above systems in some aspects: they all focus on video annotation. This system mainly focuses on the annotation part with an organized way of annotating the events found in the videos. A great addition is the analysis module. This analysis module will provide a platform to interact with the multimodal data in an intelligent way. It allows the facility to interact with the data in three levels: raw level, feature level, and semantic level. There is also an important module which is the co-analysis module. Co-analysis between multimodal data is not done in the previously mentioned tools. Hence, this module is an adequate addition to the needed annotation system. The comparison between major modules of these tools is presented in the Table 3.

Table 3: Comparison between existing tools

Tool	Video Annotation	Video Analysis	Audio Analysis	Image Analysis	Text Processing	Co-Analysis
Anvil	Yes	Yes	Yes	No	No	No
CAVA	Yes	Yes	Yes	No	No	No
ELAN	Yes	Yes	Yes	No	No	No
MRAS	Yes	No	No	No	No	No
VideoAnnEx	Yes	Yes	No	No	No	No
The Proposed framework	Yes	Yes	Yes	Yes	Yes	Yes

3 Architecture of the proposed framework

This chapter presents an architectural overview of the framework that has been implemented for use in multimodal data interaction, annotation, and analysis. The framework can be divided into two modules: annotation and analysis. The overall architecture is shown in Figure 4.

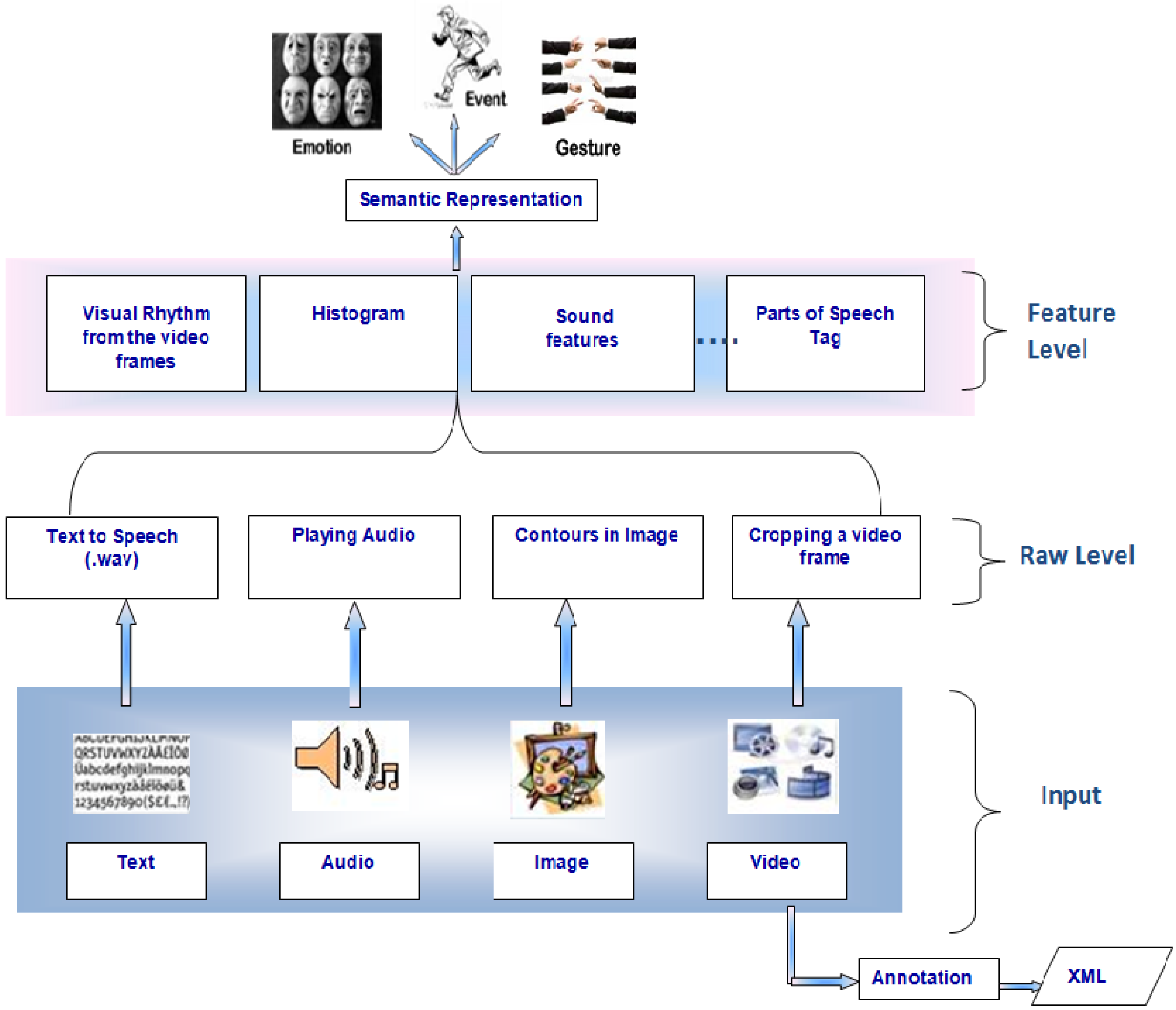


Figure 4: Basic architecture of the framework

The foremost feature of this tool is the data annotation. In this annotation module, video data annotation is included notwithstanding its complexities. Also, our research needs a lot of ground truth data collected from various video sources. For this reason, this tool can give the privilege to have our own refined, adequate, and available annotated data. This provides a major helping hand to any research interest which involves the video data.

The second module is concentrated on how the analysis of the multimodal data can be done in an interactive way. By multi-modality, we refer to these basic modes: video, image, audio, and text.

3.1 Video annotation module

In this framework, two types of annotation are proposed. One type of annotation is semantic level. In the semantic level annotation, the annotator will focus on the particular events to be noticed in a particular portion of the video. The other type of annotation is point annotation. One can play the video frame by frame and get the location information (points) of the particular objects in the video.

One of the major focuses of the framework is video annotation module that is designed for enriching the annotated video database. It is started with the key purpose that it would be served as a prerequisite or major source for the analysis of video data. It stores the proper metadata which are required for the analysis of that data. This tool ensures that the data is automatically stored. At first, the data needed to be selected from the video is categorized. The data will be annotated according to those categories.

We can derive two key points related to the process of manipulating a video document. Users tend to work with video in two ways: annotation and detailed analysis. Annotation implies ”note taking”. The annotation task is characterized by high cognitive and attentional demands. Detailed analysis typically occurs after the annotation and does not have the same constraints as annotation. In this case, the user may make many passes over a given segment of video in order to capture verbal transcriptions, behavioral interactions, and gestural or non verbal information.

3.2 Analysis module

The analysis module has these modalities: video, image, audio, and text. Every module has several necessary functionalities which will make it possible to interact with data. Regarding our research interest, specific functions are included to each modality. For example, the video module includes object tracking, body component tracking, and region locator as major functions. The image module contains many important image processing tasks such as contour tracking, good feature selection, face detection, histogram processing, and so on. For image processing tasks, various computer vision algorithms used provided by the OpenCV library [3]. The audio module includes a text-to-speech synthesis part which gives a hand to a comparative analysis between a human-uttered dialog act and an agent-uttered dialog act. In this case, the text-to-speech module performs as agent. The audio module includes a major speech feature selection from the sound files and associated analysis accordingly. The text module has two functionalities as TF (term frequency) and IDF (inverse document frequency) value creation and POS (parts of speech tagging).

Figure 5 shows the screenshot of the main window of the framework. The next two chapters will elaborately describe these two major modules: annotation and analysis, which will support the eligibility for developing this framework.

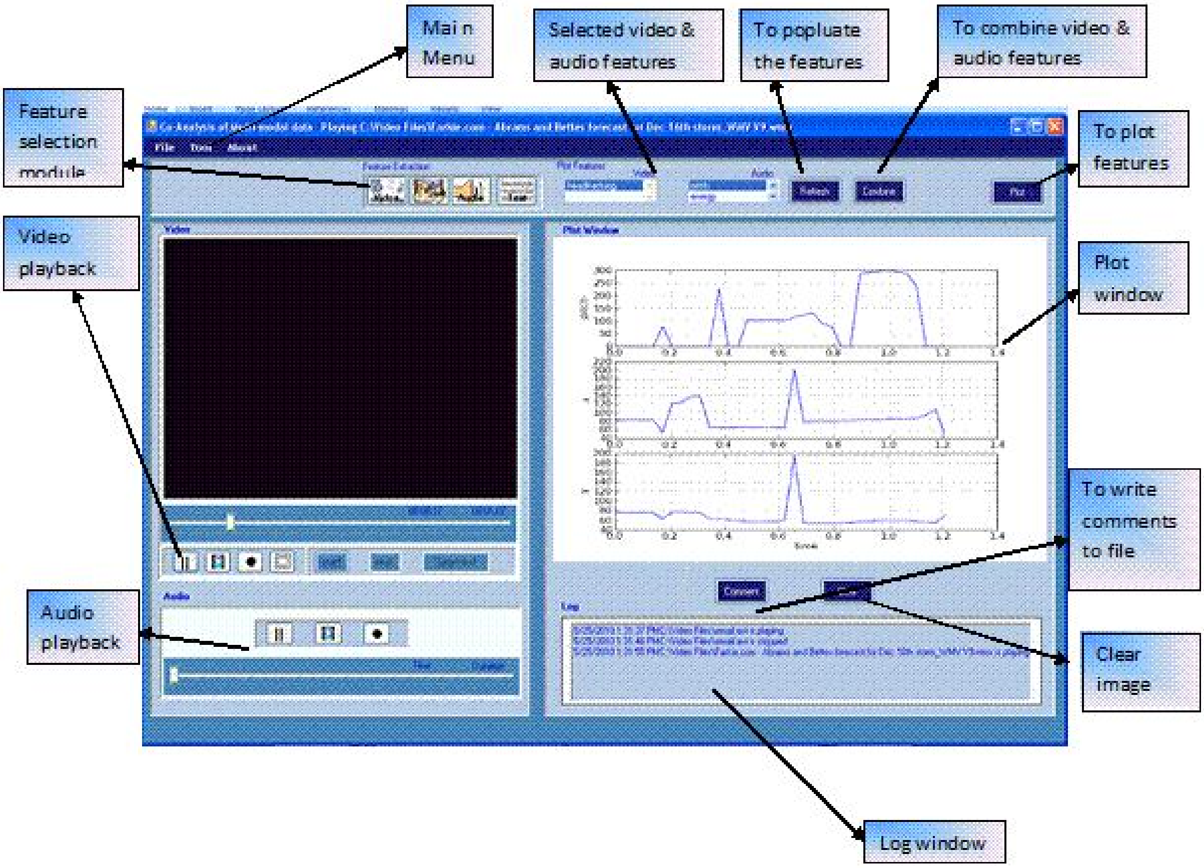


Figure 5: Main window of the framework

4 Video annotation module

This framework includes two types of annotation: semantic annotation and point annotation. Although many good video annotation frameworks exist, our aim is to set up this one so that it can be incorporated with the project's versatile features. The main goal is to have an annotated video database from which we can execute data analysis afterwards. First, the metadata is selected which is required for our analysis. From the selected meta data, we find that the frame information gives enough access to the particular video portion. Frame information and time stamps are included as the metadata. For annotation, the predefined classes are selected. Each important frame should contain associated annotation. This framework has the feature to successfully play the video frame by frame and it stores annotated data with frame information. The important feature of this module is that it is module based so that new modules can be added along with the future requirements. Since a successful video-annotation system should be provided to the users with a useful abstract of video content in a reasonable processing time, our aim is to fulfill this requirement as well.

The point annotation provides the point-wise location of the particular selected video frame. In this way, the annotator will go through the video frame by frame selecting the interesting parts of the video and the associated point location will be saved in a file. An additional functionality is that the user can select the exact portion of the whole video according to the user's interest. The particular selected portion of the video will be saved as a separate frame-image for further processing. Along with this sub-image, the particular frame number and time stamp of the specified video will be saved in an XML output file. So, in point annotation we not only have the meta data but also some high level annotations are done.

The picturesque samples of the two types of annotation: semantic and point annotation are included in Figure 6 and Figure 7 respectively.

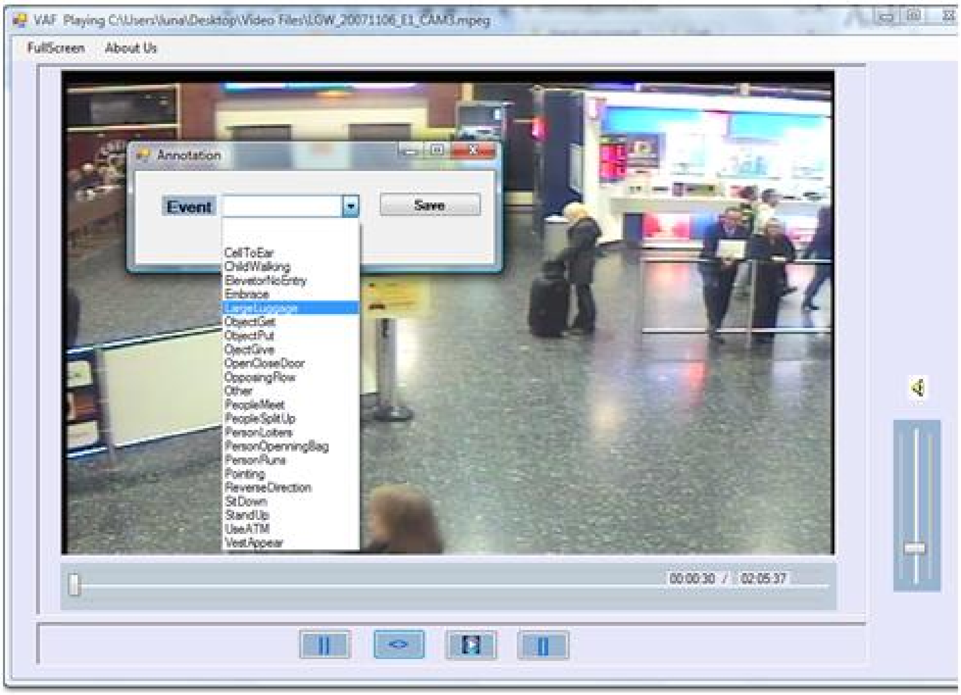
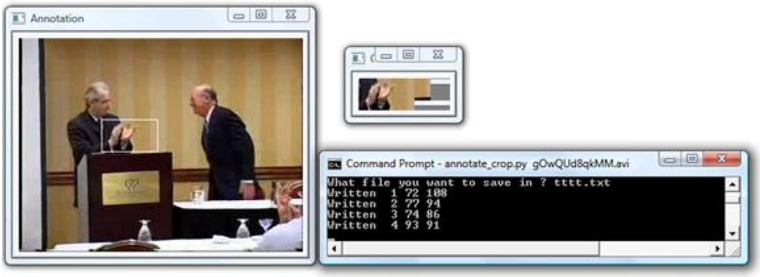


Figure 6: Semantic annotation

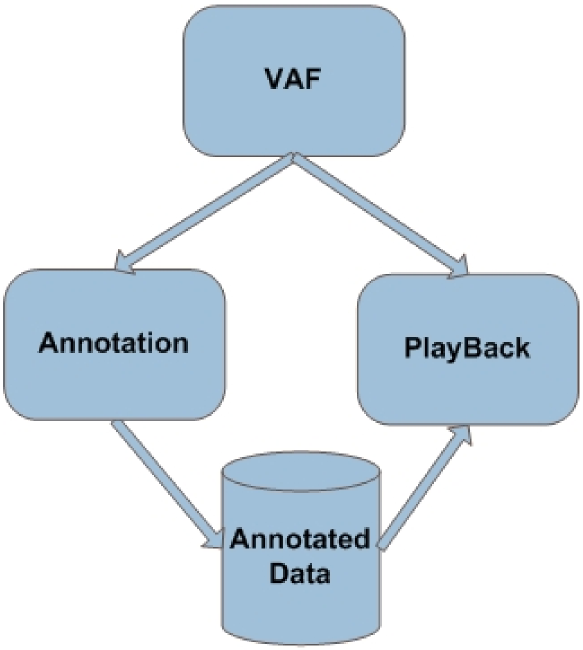


```
What file you want to save in ? tttt.txt
Written  1 72 108
Written  2 77 94
Written  3 74 86
Written  4 93 91
```

Figure 7: Point annotation

Video Annotation module has two basic parts (Figure 8).

- Annotation
- PlayBack



Basic Architecture of VAF Modules

Figure 8: Video annotation module

4.1 Annotation module

The implementation starts from a query ”how to employ a suitable metadata system for video annotation”. In this case, it is started with basic information as start and end frames and timestamps of a particular video portion containing specific event. The user also has an option to feed the list of annotations to the system. The annotation list will be added as a separate plug-in so that it can be adapted to different research premises. For instance, researchers involved in gesture recognition may populate the list with gesture primitives [12]: preparation, stroke, hold, and retraction; while as researchers involved in event detection may include walking, running, standing up and so on in the list.

4.1.1 Data structure organization

The basic data structure [13] of annotation consists of: tier, label and timeline. Annotations can have tier based format or non-tier based format. A tier is a set of annotations that share the same characteristics, e.g., one tier containing the orthographic transcription, or another tier containing the free translation. Many annotations are partitioned into a number of tiers such that each annotation is part of exactly one tier, and no two annotations within a tier overlap. These tiers are usually used to group annotations which belong to one level of analysis (e.g., verbal vs. non-verbal behavior, hand movements vs. facial expression). In non-tier annotation, the tool does not have the concept of a tier; it keeps all annotations in a single list. Annotations can consist of a single label or multiple (typed) labels for one and the same annotation. Timstamps associated with annotation could be of implicit or explicit time-line. In implicit time-line, timestamps of annotations refer directly to media times in the recording. It does not permit unspecified media offsets. It does not leave the media offsets of certain annotations unspecified.

This annotation module has the following data structure organization:

1. non-tier based formats
2. single label annotation
3. implicit timeline

Non-tier based format: The module do not have the concept of a tier; it keeps all annotations in a single list.

Single-label annotation: Annotations consist of a single label.

Implicit timeline: The timestamps of annotations refer directly to media times in the recording.

4.1.2 Methodology

At the beginning, a good platform is selected to implement this framework. In this investigation, Visual studio .NET framework 2008 is used. This has given us a much more flexibility. As we are more interested in modular based architecture, this .NET framework provides this facility.Windows Media Interface Class is manipulated for dealing with multimedia. All the necessary functions is done with MCI command string. To play the multimedia file, windows media player library is used.

The required key features of the annotation module are as follows:

1. different kinds of media files as mpeg4, mp4, avi, wmv can be played
2. pausing and playing the file frame by frame can be done
3. every selected video portion will be annotated with start and end frame and timestamps
4. every selected video event's start and end frame's information will be stored in an XML file
5. annotation categories can be fed into the system as a plug-in
6. gives the frame-rate or frame/time measurement of the played video

The Psuedocode for the Video Annoation workflow:

Algorithm 1 Semantic annotation

```
Require: Video file $V$
  Open a video file $V$
  while no end of file $V$ do
    if start of an event then
      startframe = currentframe
    end if
    if end of an event then
      endframe=currentframe
    end if
    if startframe & endframe then
      Write
    end if
  end while
```

4.1.3 Input

In this framework various kinds of video formats are incorporated. The input to this module is video file. The categories are: MPEG, MP4, WMV, and AVI.

4.1.4 Output

The output of this module is: XML and CSV (comma separated value) file. Once a video sequence undergoes the event annotation described in the previous section, the events contained within it are stored in XML file format. Since this format is easily machine readable, the analyst is able to sort through the video data much more efficiently (using the video analysis module). XML is also human readable and thus manual viewing of event-content summary may also yield a good description of the corresponding video segment. The XML document contains a reference to the video data file, video segment specific attributes (such as video length and frame rate), and data on each of the events occurring within the video segment.

A sample XML output:

```
<Events>
    < Event >
        < Starttime >00:00:57 < /Starttime >
        < Endtime >00:00:58 < /Endtime >
        < Startframe >1437 < /Startframe>
        < Endframe >1459 < /Endframe>
        < EventName >PeopleSplit < /EventName >
        < Framerate >25 < /Framerate >
    < /Event >
```

The availability of machine-readable annotation documents in XML is a big step towards the bridging of the semantic gap. A video analysis tool that takes this kind of annotation as input and organizes the corresponding video segment accordingly is certainly conceivable. This type of tool could function as an aid to a research analyst searching for ”important” events within a stream of video data.

4.1.5 PlayBack

The annotated information is saved in a specified XML file. Using Playback, we can select particular annotated information such as any specified class (e.g., video event) and from the XML file we will get the particular frame and time information. With this information the specified portion of the video can be played again for further processing.

5 Analysis module

The tool provides multimodal data analysis. The key modalities included here are: video, image, audio, and text.

5.1 Raw level interaction

The framework provides a common platform to browse all the multimodal data in one environment. Using this module, the user can do the raw level interactions with the multimodal data. The user can easily perform various tasks including loading multimodal data in a same visual display and playing a video and audio file back and forth. Moreover, it is possible to browse the video frame by frame. In addition, any image file can be loaded and then, it can be processed and can be saved to a designated folder.

5.2 Video analysis

Through this framework, the user will be able to load and play several types of videos. These are mpeg, wmv, avi, and mp4. At the time of video analysis, the user first load the video. During this time, if the user notices any event of interest then he can record the start and end of that particular event. In this way, for any important event the user can get any particular frame of interest. The user is able to grab this frame through this framework. Thus, it is possible to play around with the data in the raw level and get the feature information in one place. Think about a video that is being played; in this video, not all the visual portion has the salient feature or not all the video portion is of equal importance. Now, we are interested to select only that portion which is event of interest. This is possible in this framework. The user can click on the top-left corner of the region and bottom-right corner of the region in the video and the selected video portion can be saved as an image in a desired location for further feature processing. Various features can be extracted from the video, such as head points, hand points, etc., and visual rhythm.

5.2.1 Video segmentation

The framework provides manual video segmentation. To segment a video portion user can select start and end frame. The video is segmented within this start and end frame. This segmented video can be saved as wmv and avi format.

5.2.2 Visual rhythm

The visual rhythm [14] is a video sampling technique to represent the video by a 2D image. The key idea is to transform video into a problem of pattern detection, where each video event is transformed into a different pattern on a 2D image, called visual rhythm, obtained by a specific transformation. The visual rhythm was developed for segmenting video sequence into its components such as shot-change, cut, and dissolve.

In many research visual rhythm was used to detect cut in the video. In our research we have applied visual rhythm to transform the video into a 2D image. On these images, we have applied methods of image processing to extract the different patterns related to the events.

Let $D \subset Z^2, D = 0, ..., M - 1x0, ..., N - 1$, where M and N are the width and the height of each frame, respectively.

Definition of Frame: A frame is a function from D to Z where for each spatial position (x, y) in D, $f_t(x, y)$ represents the gray scale value of the pixel (x, y).

Definition of Video: A video V, in domain $2D + t$, can be seen as a sequence of frames f_t and can be described by

$$V = (f_t)_{t \in |0,T-1|}, \tag{1}$$

where T is the number of frames contained in the video.

Definition of Visual rhythm (Spatio-temporal slice): Let $V = (f_t)_{t \in [0,T-1]}$ be an arbitrary video, in domain $2D + t$. The visual rhythm ϑ, in domain $1D + t$, is a

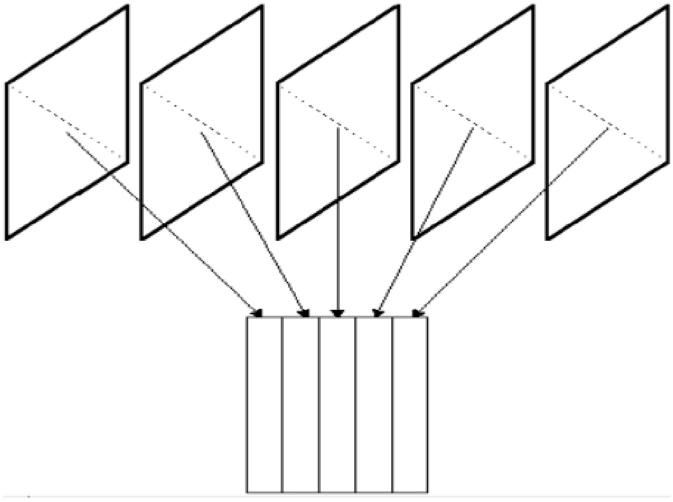


Figure 9: Visual rhythm technique

simplification of the video where each frame f_t is transformed into a vertical line on the visual rhythm that is defined by

$$\vartheta(t, z) = f_t(r_x * z + a, r_y * z + b), \tag{2}$$

where $z \in 0, ..., M_\vartheta - 1$ and $t \in 0, ..., N_\vartheta - 1$, M_ϑ and N_ϑ are the height and the width of the visual rhythm, respectively, r_x and r_y are ratios of pixel sampling, a and b are shifts on each frame. Thus, according to these parameters, different pixel samplings could be considered, for example, if $r_x = r_y = 1$ and $a = b = 0$ and $M = N$ then we obtain all pixels of the principal diagonal. If $r_x = -1$ and $r_y = 1$ and $a = M$ and $b = 0$ and $M = N$ then we obtain all pixels of the secondary diagonal. If $r_x = 0$ and $r_y = 1$ and $a = M/2$ and $b = 0$ and then we obtain all pixels of a central vertical line (Figure 10)

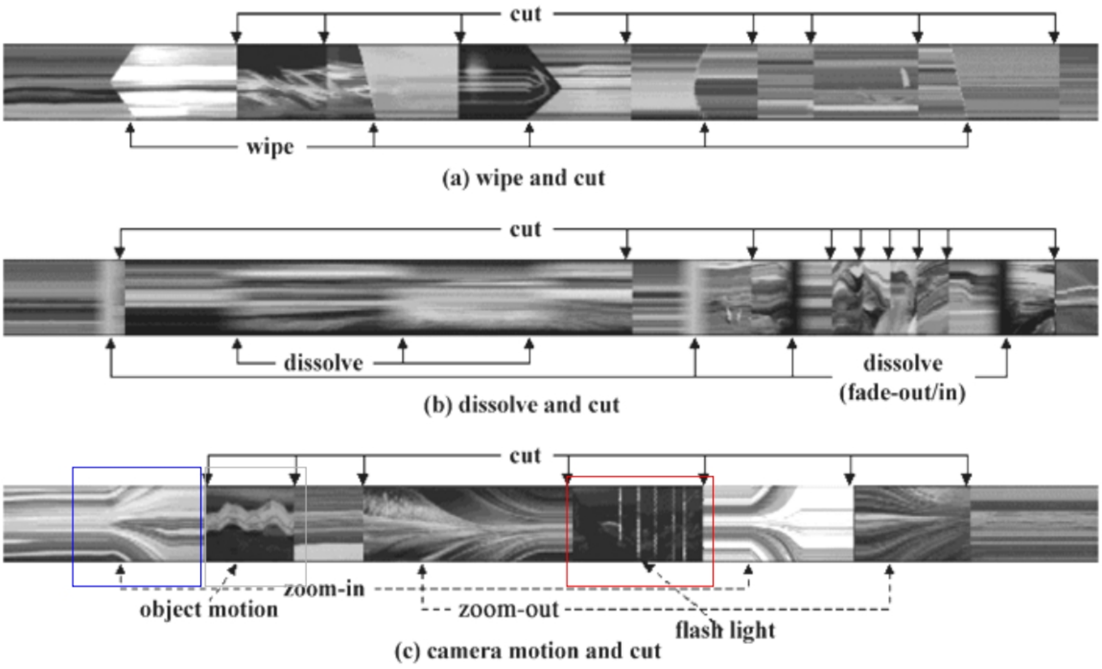


Figure 10: Example of Visual Rhythm

5.2.3 Body component tracking

This framework provides several automatic body component tracking from a video file. These are head tracking, hand tracking, upper body tracking, lower body tracking and full body tracking. The tracked points are saved in a text file with the point location information and associated frame number. Some samples of the body component tracking are given in Figure 11, Figure 12, Figure 13, and Figure 14.

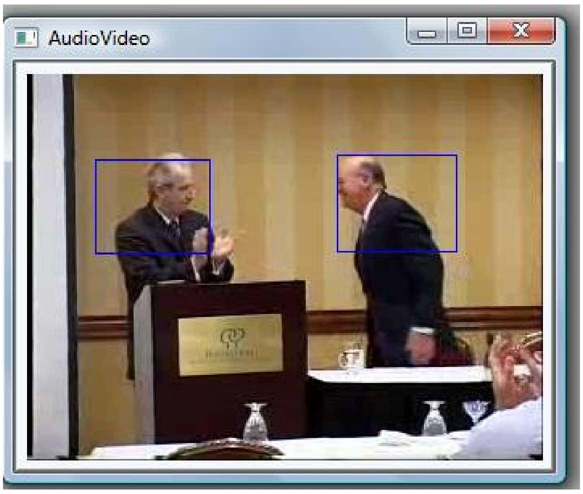


Figure 11: Upperbody tracking

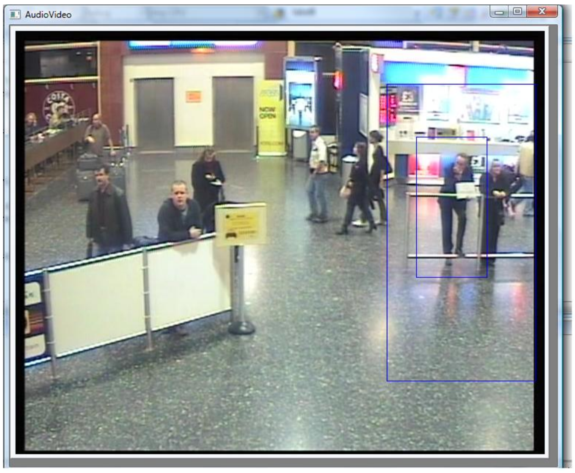


Figure 12: Fullbody tracking

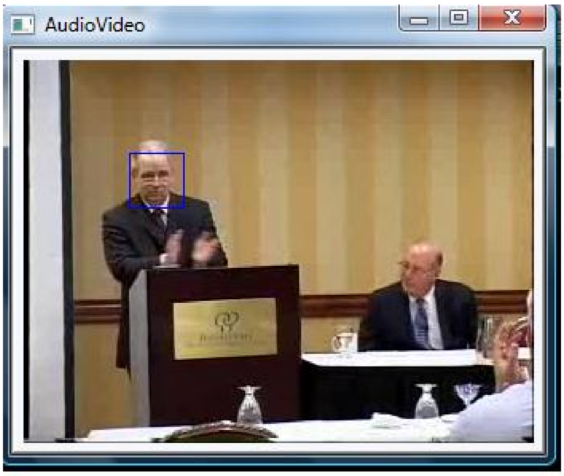


Figure 13: Head tracking

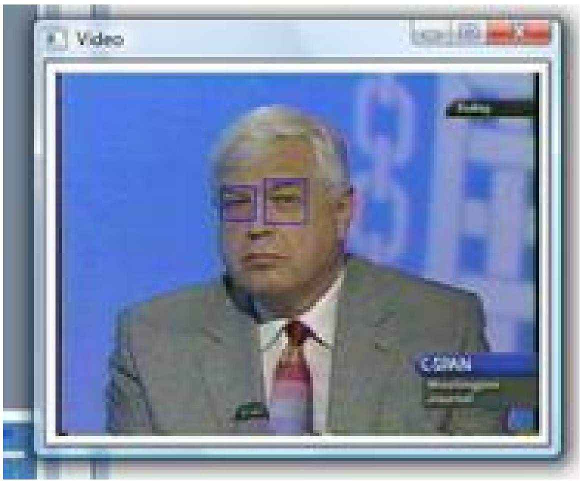


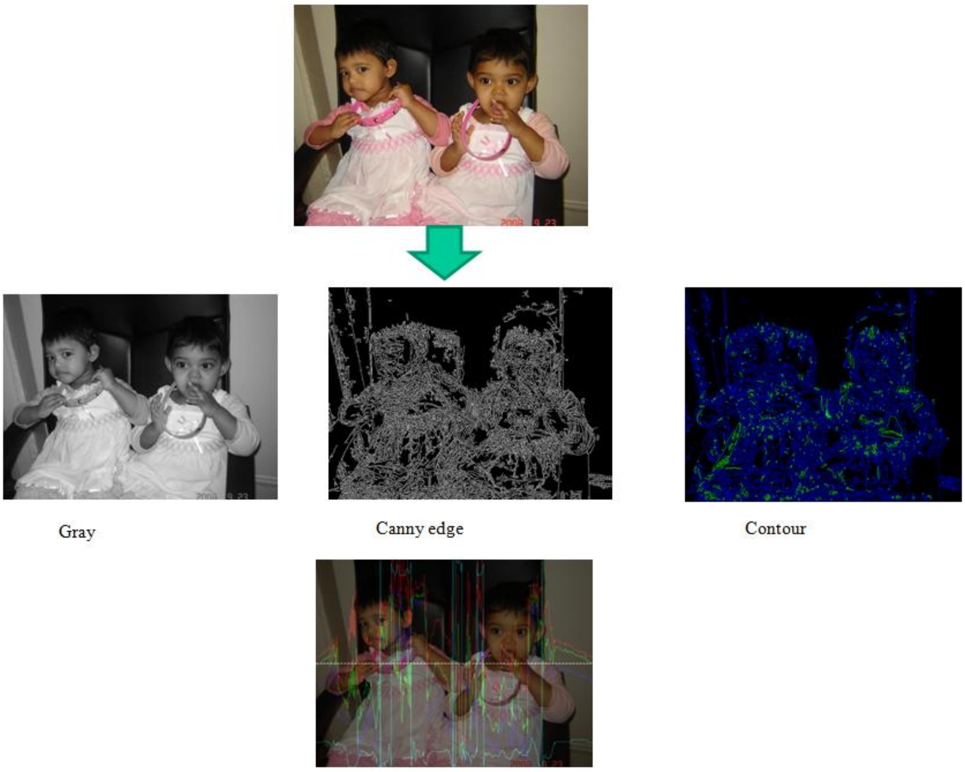
Figure 14: Eye tracking

5.3 Image analysis

In this module, we can load any image file for further processing. The loaded image can be used for some fundamental image processing. Also as raw data, histogram, and contour points can be extracted from this image. Image features are divided in four classes as shown in Table 4. In this framework, all the low level features, moments, fourier descriptor as mid level features and face detection as one of the high level features are included. In Figure 15 some examples of processed image are presented.

Table 4: Image features

Low level	**Mid level**	**High Level**	**Spectral**
edge	shape	object	spectral content
corners	texture	body part	continuous wavelet
pixel information	fourier descriptor		
gray values	moments		


Gray

Canny edge

Contour

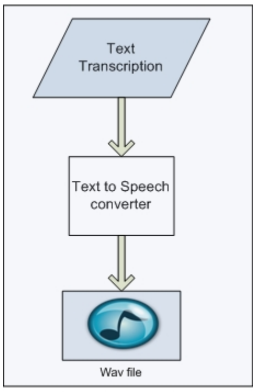
Histogram

Figure 15: Image processing

5.4 Audio analysis

This module has an important part which is text-to-speech synthesis (Figure /refText to Speech). For this synthesis, a standard voice library, Microsoft windows speech library is used. This synthesis produces a standard digital audio file from a text file. The language is for US English. This will be saved as a sound (wav) file. In many research area including dialog acts or agent-learning environment this module helps the researcher to have significant data for research purposes, for example, they can compare human-uttered dialog acts with machine-uttered dialog acts with various aspects. This module helps in the analysis of human and agent speech differences.

In the audio analysis, the framework can play sound (wav) files and user can extract features. Audio features like pitch, intensity can be extracted from both the human and agent voices. As the text-to-speech synthesis module is generating voice from text file it can be considered as an automated agent. The user can compare or co-analyze the human voice and agent voice.



Text to speech

Figure 16: Text to speech

5.4.1 Audio features

Audio has two classes of features: acoustic and prosodic feature. Pitch and intensity belongs to prosodic feature and energy and bandwidth belongs to acoustic feature.

Pitch: Sounds may be generally characterized by pitch, loudness, and quality. Pitch represents the perceived fundamental frequency of a sound. It is one of the four major auditory attributes of sounds along with loudness, timbre, and sound source location. When the actual fundamental frequency can be precisely determined through physical measurement, it may differ from the perceived pitch because of overtones, also known as upper partials, harmonic, or otherwise.

Energy: Sound energy is the energy produced by sound vibrations as they travel through a specific medium. Sound vibrations cause waves of pressure which lead to some level of compression and rarefaction in the mediums through which the sound waves travel. Sound energy is, therefore, a form of mechanical energy; it is not contained in discrete particles and is not related to any chemical change, but is purely related to the pressure its vibrations cause. Sound energy is typically not used for electrical power or for other human energy needs because the amount of energy that can be gained from sound is quite small.

Intensity: The intensity of a sound wave is the amount of power in the wave per unit area and has units of W/m2. The intensity of a sound wave depends on how far we are from a source. If we label that distance as R, then the sound intensity is

$$intensity = \frac{power}{4\pi R^2} \tag{3}$$

By definition, the intensity of any wave is the time-averaged power it transfers per area through some region of space. The traditional way to indicate the time-averaged value of a varying quantity is to enclose it in angle brackets. The unit of intensity is the watt per square meter– a unit that has no special name.

Formant: Formants are the distinguishing or meaningful frequency components of human speech and of singing. By definition, the information that humans require to distinguish between vowels can be represented purely quantitatively by the frequency content of the vowel sounds. In speech, these are the characteristic partials that identify vowels to the listener. Most of these formants are produced by tube and chamber resonance, but a few whistle tones derive from periodic collapse of Venturi effect low-pressure zones. The formant with the lowest frequency is called f1, the second f2, and the third f3.

Formants are defined as 'the spectral peaks of the sound spectrum —P(f)— of the voice. Formant is also used to mean an acoustic resonance, and in speech science and phonetics, a resonance of the human vocal tract. It is often measured as an amplitude peak in the frequency spectrum of the sound, using a spectrogram or a spectrum analyzer. In acoustics, it refers to a peak in the sound envelope and/or to a resonance in sound sources, notably musical instruments, as well as that of sound chambers.

5.5 Text analysis

In some multimodal system text files are associated with video and audio files. These are called speech transcriptions. Accordingly the transcriptions are also needed to be analyzed. This framework has some natural language processing functionalities. These are finding TF and IDF matrices and part of speech tagging.

5.5.1 TF and IDF

This module provides the TF (term-frequency)[15] from an user's input. The user will input a directory containing documents. It will provide the IDF (inverse document frequency) after creating the dictionary for the user entered word list.

In this module, the user will select a particular directory where he has the selected documents. From those documents a TF dictionary will be created. For IDF, the user will provide the words list for which the inverse document frequency is required. TF values and IDF values will be saved in the corresponding files.

Definition of TF and IDF: The tf-idf weight (term frequency-inverse document frequency) is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. The term count in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document d_j. Thus, we have the term frequency defined as follows.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{4}$$

where $n_{i,j}$ is the number of occurrences of the considered term t_i in document d_j, and the denominator is the sum of number of occurrences of all terms in document d_j.

The inverse document frequency is a measure of the general importance of the term (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$idf_i = \log \frac{|D|}{|\{d : t_i \epsilon d\}|} \tag{5}$$

with

$|D|$: total number of documents in the corpus

$|\{d : t_i \epsilon d\}|$: number of documents where the term t_i appears (that is $n_{i,j} \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to use $1 + |\{d : t_i \epsilon d\}|$.

Then

$$(tf\text{-}idf)_{i,j} = tf_{i,j} \times idf_i \tag{6}$$

A high weight in tf-idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. The tf-idf value for a term will always be greater than or equal to zero.

5.5.2 Parts of speech tagger

It is a common area for natural language processing [16]. The term natural language processing encompasses a broad set of techniques for automated generation, manipulation, and analysis of natural or human languages. Although most NLP techniques inherit largely from linguistics and artificial intelligence, they are also influenced by relatively new areas such as machine learning, computational statistics, and cognitive science. Here some very basic terminology are described briefly.

Token: Before any real processing can be done on the input text, it needs to be segmented into linguistic units such as words, punctuation, numbers, or alphanumeric. These units are known as tokens.

Tokenization: The process of splitting a sentence into its constituent tokens. For segmented languages such as English, the existence of whitespace makes tokenization relatively easy and uninteresting. However, for languages such as Chinese and Arabic, the task is more difficult since there are no explicit boundaries. Furthermore, almost all characters in such non-segmented languages can exist as one-character words by themselves, and can also join together to form multi-character words.

Sentence: An ordered sequence of tokens.

Corpus: A body of text, usually containing a large number of sentences.

Part-of-speech (POS) tag: A word can be classified into one or more lexical or part-of-speech categories such as nouns, verbs, adjectives, and articles, to name a few. A POS tag is a symbol representing such a lexical category, e.g., NN (noun), VB (verb), JJ (adjective), AT (article).

POS tagging: Given a sentence and a set of POS tags, a common language processing task is to automatically assign POS tags to each word in the sentence. For example, given the sentence, ”The ball is red,” the output of a POS tagger would be, ”The/AT ball/NN is/VB red/JJ”. Tagging text with parts-of-speech turns out to be extremely useful for more complicated NLP tasks such as parsing and machine translation.

5.6 Co-analysis of multimodal data

The hypothesis behind co-analysis of signals can be justified by taking the case where we have an audio-visual access to natural conversation. It is worth mentioning that naturally we realize there is a psychological relationship between speech and facial expressions. Hence, modeling this co-factor rather than modeling audio and video individually should be more effective for at least the following three reasons:

i) Co-analysis keeps the relevant and distinctive part of the information available from two signals which is helpful for drawing a decision boundary

ii) Co-analysis reduces dimensions (dimension will be lot less than the total dimensions needed for modeling two signals separately), hence reduces the curse of dimensionality

iii) Co-analysis reduces the possible risk of merging two different models generated from two different sources of incompatible features.

While co-analysis is supposed to be more effective in multimodal scenario, we can also use the same justification for uni-modal cases using co-analysis of multiple features or attributes (i.e., pitch and energy for speech modality).

5.6.1 Co-analysis of speech features

From the audio file, couple of features named pitch, intensity are extracted. The co-analysis is performed in these features. For example, user can select one of the audio features as pitch, and another feature as intensity. The analytical result will be displayed as graph on which correlation between pitch and intensity is visible. Thus, co- analysis can be performed using this framework. Figure 17 is an example of co-analysis between two features of an audio sample.

5.6.2 Co-analysis of video and audio

In the context of non-verbal communications, generally it is found that when a person speaks, his hands produce gesture according to the emotion and intention of the speech. Hence, this relations, if found at concurrent bases and properly modeled in a concise way, this may provide a good area of research. In this aim, we have built this co-analysis module. We can provide a way to look at the signal and sense of the non-verbal communication. Among other non-verbal communications, body posture, hand gestures are significant. So, we have a body component tracking and annotation module, which henceforth helps to define the features.

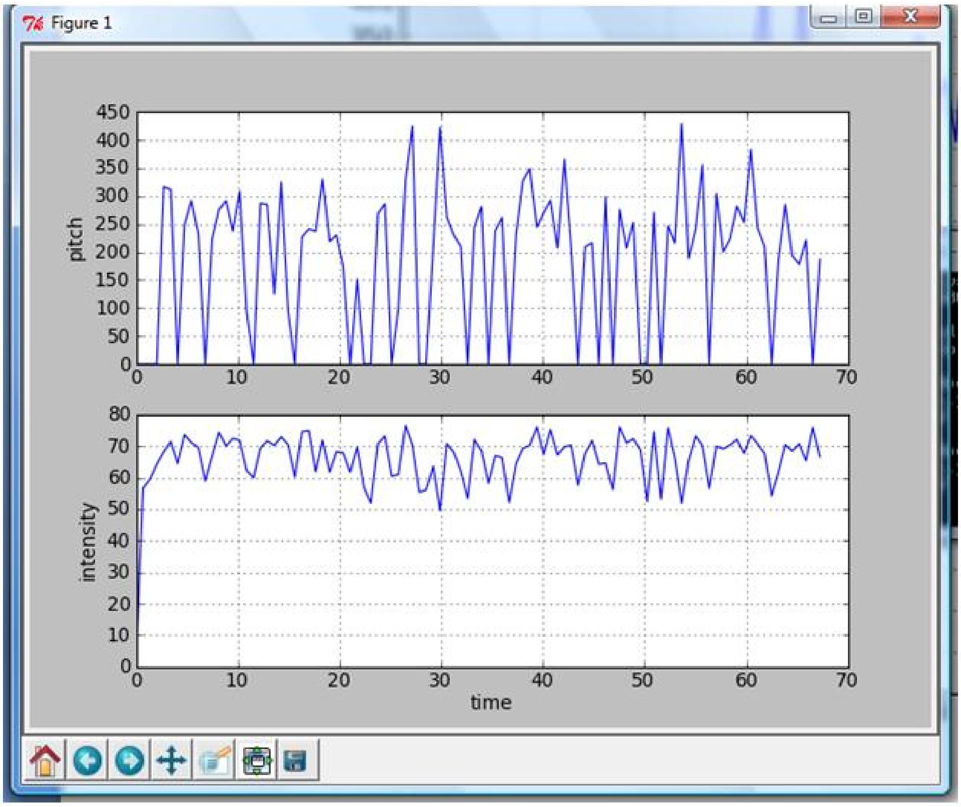


Figure 17: Co-analysis of speech features

6 Graphical User Interface and Evaluation

One of the main objective of developing this interface was to develop a user interface that will allow and even solicit human interaction and oversight, enabling effective access to the multimodal data based on principles from proven industry practices. In this framework, some of the major interface designing principles are followed.

6.1 The GUI design principles

When applied to computer software, user interface design is also known as Human-Computer Interaction or HCI. While people often think of interface design in terms of computers, it also refers to many products where the user interacts with controls or displays. Optimized user interface design requires a systematic approach to the design process. But, to ensure optimum performance, usability testing is required. This empirical testing permits naive users to provide data about what does work as anticipated and what does not work.

In order to design usable interfaces, it is necessary to understand the process by which users perform actions. An action is defined as a collection of simple behaviors performed in sequence to complete a very small portion of the larger task a software application facilitates. Actions may be divided into seven stages [17]: forming the goal, forming the intention, specifying an action, executing the action, perceiving the state of the world, interpreting the state of the world, and evaluating the outcome.

Usability Guidelines:

Good conceptual model:

In order to form goals and intentions and specify actions, the user depends on a conceptual model, a mental representation of the application's available functions and of the actions the user must perform to utilize those functions. Good design provides users with good conceptual models. A good conceptual model requires an accurate sense of the application's mapping of controls (command buttons, menu items, scroll bars, etc.) to functions. To provide this understanding, an application's controls should map to functions in accordance with users' intuition, there should be as close as possible to a one-to-one correspondence between controls and functions, and the application must offer clear, visible cues as to the functions of each of its controls.

Visibility:

Appropriately chosen graphical images often make better visual cues than text because they take less time to interpret, form a stronger impression on memory, and contribute to users' motivation. An application's graphics should be stylistically consistent. Simpler graphics are easier to learn to recognize than complex graphics, as are graphics which metaphorically represent the functions of the controls they mark compared against abstract logos[18].

Mapped Control:

Providing one function per control facilitates graphical identification of controls, but it also makes controls easier to use and functions easier to remember. If a control has multiple functions, some of its functions may be difficult to discover, but a visible control with one function automatically reminds the user of its function simply by being visible. Multiple functions per control are likely to result in confusion and frustration[17]. Functions that map arbitrarily to controls take longer to learn and when their misuse results in error, it is more difficult to determine the cause of the error and correct it.

Memory and feedback:

Long term memory retention is also enhanced by association. New information is learned better when it can be integrated with preexisting knowledge; thus, the user can more readily learn to use controls that function as the user might expect them to upon inspection. The software designer can achieve this by mapping controls in accordance with physical analogies and standardized practices. Sliding bars, for example, should increase values as the user slides them upward or to the right. Visual cues are an effective means toward providing feedback and should be utilized according to the same principles of clarity and organization that guide their usage as control markers. Auditory feedback may be useful in certain applications, but its use should be minimized since many users find it annoying, it demands much of the user's attention, and may be considered intrusive by those in close proximity to the user. Users should be able to undo actions whenever possible. When it will not be possible to undo an action, the user must first receive a clear, thorough warning about the action's consequences and have the opportunity to cancel the action.

A usable application, therefore, provides users with an accurate conceptual map through the use of clear, complete, immediate feedback and intuitive, one-to-one mapping of well organized, visibly-identified controls to functions; however, these necessary principles are insufficient given that a diversity of users implies a diversity of interactive expectations and capabilities.

UI principles and psychology

In the context of psychology [19], five human functions can be related with the principles of user interface. They are movement, perception, language, memory, and thinking.

Movement means how efficient user input on an application; perception means steering attention and recognizing information; language means clear use of language; memory means learning and memory; and thinking means mental load. Details with examples are given as follows.

Movement

Large buttons: Larger buttons are easier to hit and therefore cost less effort than clicking a small button.

Less input: When one let the user select an item from a list, it is better to think about using a default selection, usage dependant groupings (recently, frequently, hardly, never used) and keying and/or pointing possibilities. User input has to be as easy for the user as possible.

Perception

This indicates how we use our eyes in our daily life, by looking and reading, determine what interfaces should look like. Perception includes size, form, luminance, color, blinking, where to place information, tables.

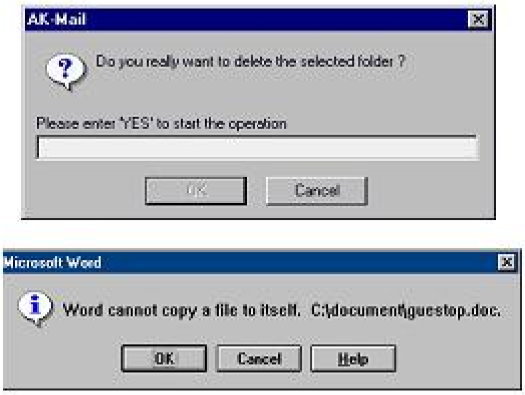
Eye fixation: When one fix one's eyes on a certain point; one can sharply see the things on that point. The direct area around the fixation point already gets a little blur. The further away from the fixation point, the blurrier it gets. So in order to read a line of words, looking at the center of the line will not work. One has to move eyes over the text to be able to read it.

Color: The visibility of colors has much to do with luminance, color-blindness and biology. For example: it is better not to use saturated colors, because the intensity of these colors is too high. The eyes have to make effort not to fixate on these colors alone. It is better not to use color for quantitative information. Gradients of luminance are very suitable for displaying quantitative information. It is better not to use color for unknown meaning. When you intend to use a color, always think what its function will be and equally important: if the user understands this.

When placing elements, one shouldn't concentrate on the distance from the page/screen margin, but one should concentrate on the distance from related elements. With input fields in forms, the labels should be kept as close to the input field as possible (again within one eye fixation). When label and input field are too distant from each other, the possibility exists that the user identifies the input field with the wrong label.

Language

The language should be of less words, clear words and crisp sentences. As in our GUI, we use many dialog forms for example error messages; we should take care of the language in those. An example is shown in Figure 18.



AK-Mail

Do you really want to delete the selected folder ?

Please enter 'YES' to start the operation

Microsoft Word

Word cannot copy a file to itself. C:\document\guestop.doc.

Figure 18: Error messages

Memory

Being consistent in the language positively affects the clarity of the interface. The less consistent the language is, the more one increase the mental load of the user, because he has to remember multiple terms/objects for one element. The GUI should have consistent input keys, presentations, abbreviations, program standards and platform standards.

Help: A tool tip (Figure 19) has an unobtrusive presentation, a verb and a noun, the correct expert level, concise text, and a correct time delay (0.5 seconds).

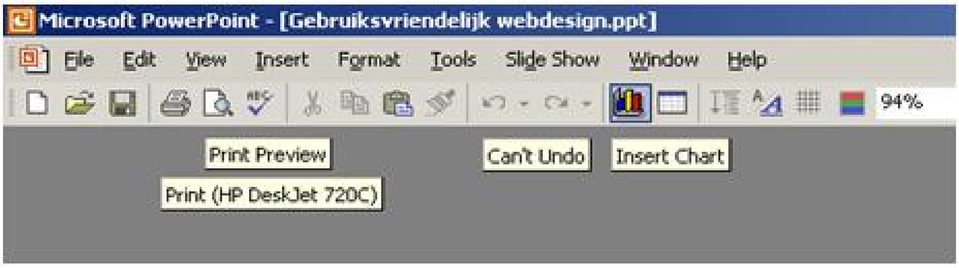

Microsoft PowerPoint - [Gebruiksvriendelijk webdesign.ppt]
File Edit View Insert Format Tools Slide Show Window Help
94%
Print Preview
Print (HP DeskJet 720C)
Can't Undo
Insert Chart

Figure 19: Tooltip example

GUI of the framework

In the current framework, most of the design principles stated above are followed as much as possible in the areas of visibility, color, luminance, language, consistency, memory, and help. The proper dialog boxes are used appropriately as needed. For the user’s help, the tooltip is used. The overall interface has given a standard size and color.

6.2 Usability

Usability [20] applies to all aspects of a system with which a human might interact, including installation and maintenance procedures. It is important to realize that usability is not a single, one-dimensional property of a user interface. Usability has multiple components and is traditionally associated with these five usability attributes:

Learnability: the system should be easy to learn so that the user can rapidly start getting some work done with system.

Efficiency: the system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.

Memorability: the system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.

Errors: the system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them, further, catastrophic errors must not occur.

Satisfaction: the system should be pleasant to use, so that users are subjectively satisfied when using it.

With these abstract concept of ”usability” in terms of these more precise measurable components, we arrive at an engineering discipline where usability is not just argued about but is systematically approached, improved, and evaluated.

Usability is typically measured by having a number of test users use the system to perform a specified set of tasks, though it can also be measured by having real users in the field perform whatever tasks they are doing anyway. In either case, an important point is that usability is measured relative to certain users and certain tasks. It could well be the case that the same system would be measured as having different usability characteristic if used by different users for different tasks.

To determine a system’s overall usability on the basis of a set of usability measures, one normally takes the mean value of each of the attributes that have been measured and checks whether these means are better than some previously specified minimum. Since users are known to be very different, it is probably better to consider the entire distribution of usability measures and not just the mean value. For example, a criterion for subjective satisfaction might be that the mean value should be at least 4 on a 1-5 scale; that at least 50% of the users should have given the system the top rating, 5; and that no more than 5% of the users gave the system the bottom rating, 1.

Learnability

Learnability is one of the most fundamental usability attributes, since most systems need to be easy to learn, and since the first experience most people have with a new system is that of learning to use it. Certainly, there are some systems for which one can afford to train users extensively to overcome a hard-to-learn interface, but in most cases, systems need to be easy to learn. One simply picks some users who have not used the system before and measures the time it takes them to reach a specified level of proficiency in using it. The test users should be representative of the intended users of the system, and there might be a need to collect separate measurements from complete novices without any prior computer experience.

Efficiency of use

To measure efficiency of use for experienced users, one obviously needs access to experienced users. For systems that have been in use for some time, ”experience” is often defined somewhat informally, and users are considered experience either if they say so themselves or if they have been users for than certain amount of time, such as a year. Experience can also be defined more formally interims of number of hours spent using the system, and that definition is often used in experience with new systems without an established user’s base: test users brought in and asked to use the system for a certain number of hours, after which their efficiency is measured.

A typical way to measure efficiency of use is thus to decide on some definition of expertise, to get a representative sample of users with that expertise, and to measure the time it takes these users to perform some typical test tasks.

Memorability

Casual users are the third major category of users besides novice and expert users. Casual users are people who are using a system intermittently rather than having the fairly frequent use assumed for expert users. However, in contrast to novice users, casual users have used a system before, so they do not need to learn it from scratch, they just need to remember how to use it based on their previous learning.

Having an interface that is easy to remember is also important for users who for some reason have temporarily stopped using a program. To a great extent, improvements in learnability often also make an interface easy to remember, but in principle, the usability of returning to a system is different from that of facing it for the first time.

Few and noncatastrophic errors

Users should make as few errors as possible when using a computer system. Typically, an error is defined as any action that does not accomplish the desired goal, and the system's error rate is measured by counting the number of such actions made by users while performing some specified task. Error rates can thus be measured as part of an experiment to measure other usability attributes.

Subjective satisfaction

The final usability attribute, subjective satisfaction, refers to how pleasant it is to use the system. Subjective satisfaction may be measured by simply asking the users for their subjective opinion. From the perspective of any single user, the replies to such a question are subjective, but when replies from multiple users are averaged together, the result is an objective measure of the system's pleasantness. Since the entire purpose of having a subjection satisfaction usability attribute is to assess whether users like the system, it seems highly appropriate to measure it by asking the users.

Likert scale

In survey questionnaires, users are typically asked to rate the system on 1-5 or 1-7 rating scales that are normally either Likert [21] scales or semantic different scales. For a Likert scale, the questionnaire postulates some statements and asks the users to rate their degree of agreement with the statement. When using a 1-5 rating scale, the reply options are typically 1=strongly disagree, 2= partly disagree, 3= neutral, 4= agree, and 5= strongly agree. A final rating for the measure is often calculated simply as the mean of the ratings for the individual answers.

6.3 Survey and evaluation

This chapter presents the performance evaluation given by a reasonable number of diversified users through collecting their initial reviews with the first phase of the tool. To evaluate the framework's usability and for data collection, we ran experiments with about 10 graduate students. We used several video contents edited into short segments from the videos collected from various legitimate sources. Content included news, surveillance video, presentation/lecture, and variety. Then, we conducted a survey on the usability of the system.

Research Questions: The aim of this study is investigated through the following questions:

1. To what extent this Interaction and Annotation Tool impose mental load?

2. Does this mental load affect the overall satisfaction of the usability of the tool?

Participants: In this study, 10 students (active researchers) were selected from the department of Electrical and Computer Engineering, The University of Memphis. The fact that all the students are involved in different research activities including from bioinformatics, medical imaging, gesture recognition, and multimodal annotation provided a variety of opinions about the usage of this tool.

Data Collection Procedures: The questionnaire was adapted to a 5-point scale ranging from 'Strongly Agree' to 'Strongly Disagree' and they are coded as (Strongly Agree =5, Agree=4, Neutral=3, Disagree=2, Strongly Disagree=1). The purpose and different terms of the questionnaire were explained to the test users by the developer. Students were informed that the information they gave would be kept confidential and be used for research purposes only. Followings are the variables that were investigated:

1. Overall qualitative: On this scale, there are 3 items (items 1-3, see Appendix A.1) that would show the overall qualitative measurement of the tool.

2. Usability: This scale includes 4 items (items 4-7, see Appendix A.1) and the respondents are asked regarding the usability of the tool.

3. Mental Effort: This scale includes 3 items (items 8-10, see Appendix A.1) and the respondents are asked regarding their mental load of while using the tool. Also 4 open-ended questions were constructed to elicit qualitative information to check whether there are any future changes or modifications in the tool.

A second survey is conducted to obtain the details of usability. For this the following variables were investigated.

1. Learnability: On this scale, there are 2 items (items 1-2, see Appendix A.2) that would show the learnability aspect of the tool.

2. Efficiency: On this scale, there are 2 items (items 3-4, see Appendix A.2) that would show the efficiency of the tool.

3. Memorability: This scale includes 2 items (items 5-6, see Appendix A.2) and the respondents are asked regarding the memorability of the tool.

4. Errors: This scale includes 3 items (items 7-9, see Appendix A.2) and the respondents are asked regarding the errors feature of the tool.

5. Satisfaction: This scale includes 3 items (items 10-12, see Appendix A.2) and the respondents are asked regarding the satisfaction aspect of the tool.

Data Calculation Procedures: The data was calculated using Microsoft Excel. Descriptive statistics (mean, frequency, and standard deviation, and standard error) were carried out for all items involved in this study (Table 2, see Appendix).

6.3.1 Results

The questionnaire survey was comprised of three sections: overall qualitative, mental load, and usability. The sections used 5-point Likert scales from strongly disagree to strongly agree.

The mean score of the questions in overall qualitative measure illustrated that students agreed to the overall quality of the tool. The mean score is equal to agree (4), hence it indicates that the quality is considered to be acceptable. The other mean scores show that usability is nearly equal to agree (4), this also indicates a better result. Above all, the mean score of mental effort which shows the mental load of the users while using the tool is quite close to value of neutral. It is required to have this value near to disagree (2). Among the three mean scores, mental effort has the minimum which is acceptable. Yet, this value should be decreased to the level of disagree (2). Here, it is seen that although the mental effort is near to neutral it does not affect the acceptance of overall quality of the tool. It is also important to mention here, that the students had a variety of backgrounds, some with strong domain knowledge. Therefore, as it is shown in the Figure 20, the standard error is significantly large for mental effort. This implies that, student with domain knowledge may have better understanding of the usability of the software and therefore, will likely be remembering the different functionalities. Students with limited background knowledge will likely have more difficulty because they are also new to the concepts presented before them. Therefore, the average score for mental effort is an overestimate. We except the real users of this tool to have some background knowledge of the topic.

The details of data of the two surveys are shown in Figure 21 and Figure 23, and the comparisons of their measures are shown in Figure 20 and Figure 22.

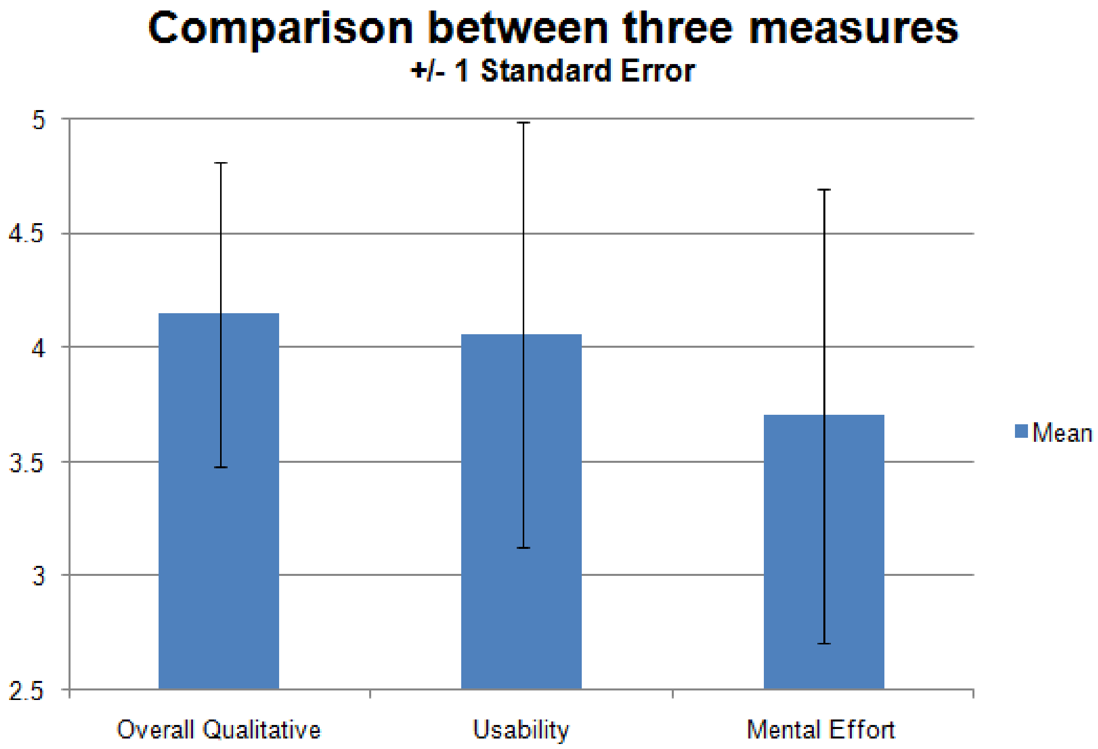


Figure 20: Comparison between the measures

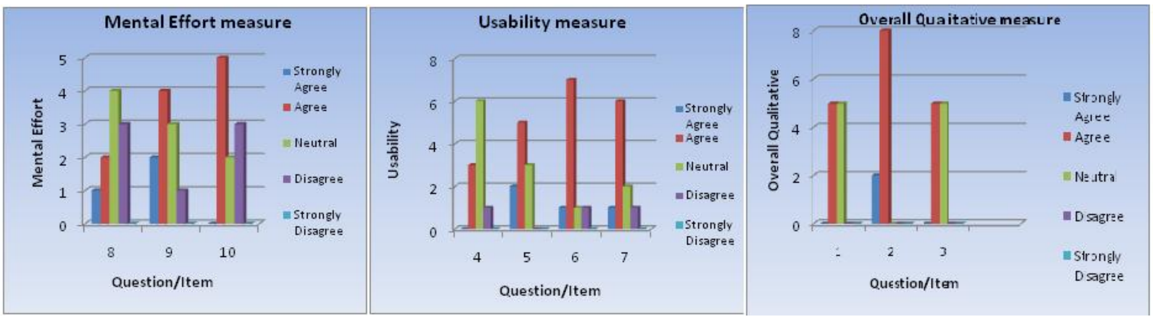


Figure 21: Three features of the tool

The second questionnaire survey is comprised of five measures of usability: learnability, efficiency, memorability, errors and satisfaction. The sections used five-point Likert scales from strongly disagree to strongly agree.

A close examination of the usability aspect of the tool is found by the second questionnaire survey. The comparison between the mean scores shows that satisfaction of the tool is quite agreeable to all the users instead of low measure in memorability and errors category. Yet, if the errors of the tool can be handled more efficiently it would give a better overall usability score of the tool.

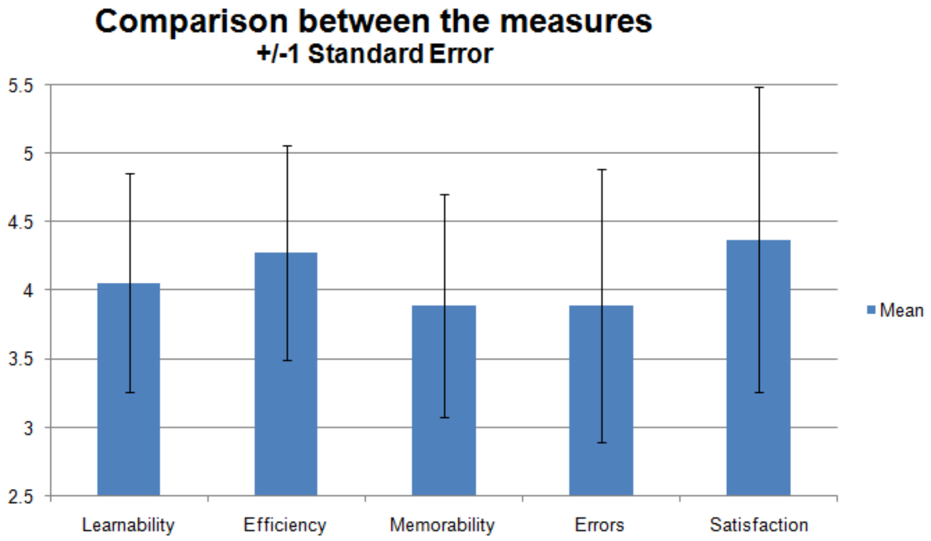


Figure 22: Comparison between usability measures

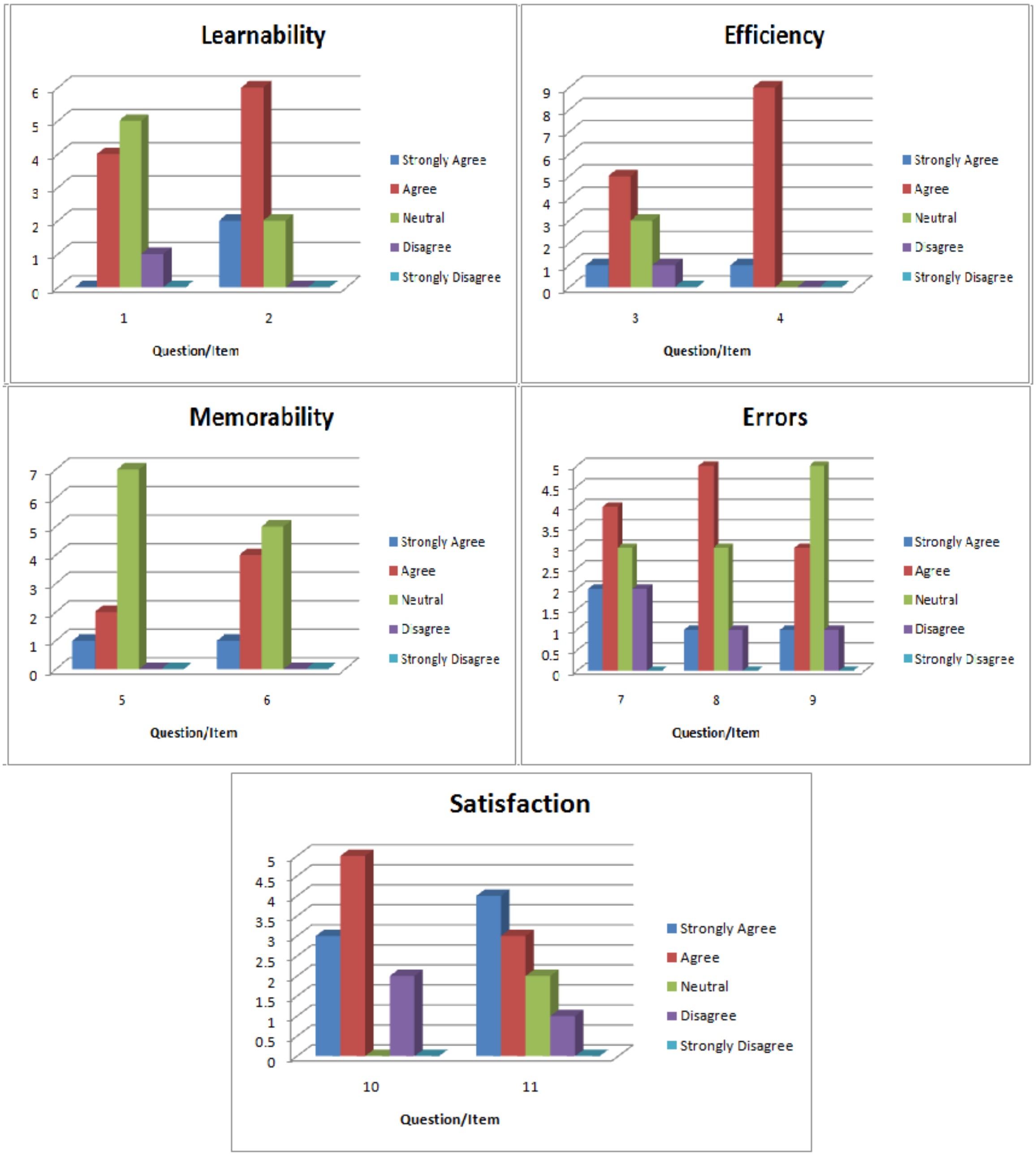


Figure 23: Five measures of Usability

7 Conclusion

The focuses of this research encompass several aspects: interaction, annotation and analysis of multimodal data. Consequently, an innovative tool is created that provides an integrated architecture enabling researchers to have access to multimodal data. This thesis proposes an integrated framework which will include the multimodal data interactions and annotation in a single platform. In this course, existing tools and standards of annotation has been reviewed and a new module has been added which will give the convenience to perform the co-analysis of multimodal data.

7.1 Impact

Researchers involving video event analysis and gesture recognition, in CVPIA lab (computer vision perception and image analysis lab, the University of Memphis) has been using this annotation and analysis tool. The software can be extended through a plug-in-loading mechanism. It is also one of the few tools that provide a co-analysis module. The key challenge is to integrate codes (computer programs) written using different programming languages and platforms, displaying the results and multimodal data in one platform. This fully integrated tool achieved the stated goals and objective. It is a valuable addition to the list of very few existing tools that are useful for interaction, annotation and analysis of multimodal data.

7.2 Future work

There has been a couple of drawbacks in this tool such as long processing time for extraction of some features, dealing with different video data formats, etc.The software has gone through iterative refinement with input from the researchers involving with video, audio, and speech. More modules will be attached according to the updated research needs. Since, it involves the integration of couple of different platforms, we are trying to resolve the complexity and make it more faster and robust.

Acknowledgment

This research was partially supported by grant NSF-IIS-0746790. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding institution.

References

[1] M. Bordegoni, G. Faconti, S. Feiner, M. T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson, "A standard reference model for intelligent multimedia presentation systems," 1997.

[2] N. O. Bernsen, "Defining a taxonomy of output modalities from an hci perspective," *Comput. Stand. Interfaces*, vol. 18, no. 6-7, pp. 537–553, 1997.

[3] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[4] C. yung Lin, B. L. Tseng, and J. R. Smith, "Video collaborative annotation forum: Establishing ground-truth labels on large multimedia datasets," in *In Proceedings of the TRECVID 2003 Workshop*, 2003.

[5] M. Kipp, "Anvil - a generic annotation tool for multimodal dialogue," 2001.

[6] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot International*, vol. 5, no. 9/10, pp. 341–345, 2001.

[7] M. Kipp, "Spatiotemporal coding in anvil," in *Proceedings ofthe 6th international conference on Language Resources and Evaluation (LREC-08)*, 2008.

[8] A. Bargeron, D.and Gupta, J. Grudin, E. Sanocki, and F. Li, "Asynchronous collaboration around multimedia and its application to on-demand training," in *Proceedings of the 34th Hawaii International Conference onSystem Sciences (HICSS-34)*, (Maui, Hawaii), Institute of Electrical and Electronics Engineers, Inc., IEEE, January 3-6 2001.

[9] H. Brugman, A. Russel, and X. Nijmegen, "Annotating multi-media / multimodal resources with elan," in *In proceedings of LREC*, pp. 2065–2068, 2004.

[10] D. Yamamoto and K. Nagao, "ivas: Web-based video annotation system and its applications," in *3rd International Semantic Web Conference(ISWC2004)*, 2004. Available: `http://iswc2004.semanticweb.org/demos/29/paper.pdf`.

[11] http://www.mpi.nl/world/tg/CAVA/CAVA.html.

[12] S. Kettebekov, M. Yeasin, and R. Sharma, "Prosody based co-analysis for continuous recognition of coverbal gestures," in *Proc. ICMI02*, pp. 161–166, 2002.

[13] T. Schmidt, S. Duncan, O. Ehmer, J. Hoyt, M. Kipp, D. Loehr, M. Magnusson, T. Rose, and H. Sloetjes, "An exchange format for multimodal annotations," pp. 207–221, 2009.

[14] S. J. Ferzoli, S. Jamil, F. G. Aes, M. Couprie, A. De, and A. A. Ujo, "A method for cut detection based on visual rhythm," unpublished manuscript.

[15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513 – 523, 1988.

[16] H. V. Halteren, J. Zavrel, and W. Daelemans, “Improving accuracy in wordclass tagging through combination of machine learning systems,” *Computational Linguistics*, vol. 27, 2001.

[17] D. A. Norman, “The design of everyday things,” 2002.

[18] S. Watzman, “Visual design principles for usable interfaces,” pp. 263–285, 2003.

[19] F. d. Dopper, “The notes from a workshop on interaction design and psychology by leonard verhoff,” 2002.

[20] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995.

[21] N. Cliff and J. Long, “Note j: Ordinal, nominal and likert scale variables ordinal issues 9/4/07 to 9/9/07 on sci.stat.edu,” unpublished manuscript.

Appendix

A Questionnaire set

A.1 Questionnaire set 1

Performance Evaluation Questionnaire

Write your own review on Interaction and Annotation Framework.Please put a check mark on the desired answer in the following questions. Completing a review does not require your personal information, and you will not be contacted regarding your review. For questions about this framework, please contact aahmed1@memphis.edu.

1. Considering all aspects of the experience, you are satisfied with this Interaction and Annotation Framework.

Strongly agree Agree Neutral Disagree Strongly disagree

2. Interaction and Annotation Framework has the functions and features to perform as expected.

Strongly agree Agree Neutral Disagree Strongly disagree

3. The layout of this interface is satisfactory.

Strongly agree Agree Neutral Disagree Strongly disagree

4. Interaction and Annotation Framework is easy to use.

Strongly agree Agree Neutral Disagree Strongly disagree

5. Interaction and Annotation Framework has few frequencies of serious errors.

Strongly agree Agree Neutral Disagree Strongly disagree

6. You need little effort to navigate this framework (e.g., finding your way around).

Strongly agree Agree Neutral Disagree Strongly disagree

7. The contents of the interface are well organized.

Strongly agree Agree Neutral Disagree Strongly disagree

8. You need little mental and physical activity to use this framework (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.).

Strongly agree Agree Neutral Disagree Strongly disagree

9. Interaction and Annotation Framework executes its functions with moderate response time or speed.

Strongly agree Agree Neutral Disagree Strongly disagree

10. You feel little stress (discourage, irritation, annoyance) during the use of Interaction and Annotation Framework. The information layout and locations are consistent when following instructions.

Strongly agree Agree Neutral Disagree Strongly disagree

A.2 Questionnaire set 2

Performance Evaluation Questionnaire

Write your own review on Interaction and Annotation Framework.Please put a check mark on the desired answer in the following questions. Completing a review does not require your personal information, and you will not be contacted regarding your review. For questions about this framework, please contact aahmed1@memphis.edu.

1. This tool is easy to use as a first time user.

Strongly agree Agree Neutral Disagree Strongly disagree

2. This tool has the functions and features to perform as expected.

Strongly agree Agree Neutral Disagree Strongly disagree

3. After getting started, you can easily comprehend the tasks you want to do.

Strongly agree Agree Neutral Disagree Strongly disagree

4. This tool executes its functions with moderate response time or speed.

Strongly agree Agree Neutral Disagree Strongly disagree

5. You need lesser effort to navigate this framework (e.g. finding your way around).

Strongly agree Agree Neutral Disagree Strongly disagree

6. The interface layout is organized as self-informative.

Strongly agree Agree Neutral Disagree Strongly disagree

7. This tool has few frequencies of serious errors.

Strongly agree Agree Neutral Disagree Strongly disagree

8. The error messages or alerts are informative enough to take the right step.

Strongly agree Agree Neutral Disagree Strongly disagree

9. The unexpected errors are properly handled by suitable error alerts.

Strongly agree Agree Neutral Disagree Strongly disagree

10. You think that the information layout and locations are consistent through the instruction.

Strongly agree Agree Neutral Disagree Strongly disagree

11. The contents of the interface are well organized.

Strongly agree Agree Neutral Disagree Strongly disagree

12. You are satisfied with the accuracy and time it take to find information (availability of help manual) for this tool.

Strongly agree Agree Neutral Disagree Strongly disagree

A.3 Data table

Table 5: Data table: survey 1

Q No.	S A(5)	A(4)	N(3)	D(2)	S D(1)	Mean	Standard Deviation
1	0	5	5	0	0	3.88	1.97
2	2	8	0	0	0	4.66	2.16
2	2	8	0	0	0	4.66	2.16
3	0	5	5	0	0	3.88	1.97
4	0	3	6	1	0	3.55	1.88
5	2	5	3	0	0	4.33	2.08
6	1	7	1	1	0	4.22	2.05
7	1	6	2	1	0	4.11	2.02
8	1	2	4	3	0	3.44	1.85
9	2	4	3	1	0	4.11	2.02
10	0	5	2	3	0	3.55	1.88

Table 6: Data table: survey 2

Q No.	S A(5)	A(4)	N(3)	D(2)	S D(1)	Mean	Standard Deviation
1	0	4	5	1	0	3.66	0.60
2	2	6	2	0	0	4.44	0.66
3	1	5	3	1	0	4.00	0.88
4	1	9	0	0	0	4.55	0.33
5	1	2	7	0	0	3.77	0.64
6	1	4	5	0	0	4.00	0.66
7	2	4	3	2	0	3.90	1.21
8	1	5	3	1	0	4.00	0.88
9	1	3	5	1	0	3.77	0.86
10	3	5	0	2	0	4.33	1.41
11	4	3	2	1	0	4.44	1.33
12	2	6	1	1	0	4.33	0.97