FIDGET DETECTION FOR AUDIO VIDEO MEETING ANALYSIS

A Thesis

Presented to

the Graduate School of

Clemson University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Electrical Engineering

---

by

Prashant Oswal

August 2006

Advisor: Dr. Stanley T. Birchfield

August 4, 2006

To the Graduate School:

This thesis entitled “Fidget Detection For Audio Video Meeting Analysis” and written by Prashant Oswal is presented to the Graduate School of Clemson University. I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science with a major in Electrical Engineering.

Dr. Stanley T. Birchfield, Advisor

We have reviewed this thesis
and recommend its acceptance:

Dr. John N. Gowdy

Dr. Stephen J. Hubbard

Accepted for the Graduate School:

ABSTRACT

This work presents the initial design and implementation of a meeting analysis system built using simple off-the-shelf microphones and webcameras. The main contribution is the development of a novel approach to video analysis, which we call *fidget detection*. All the current meeting analysis systems use background subtraction as the first stage for video analysis. Background subtraction is used on the assumption that meeting participants will enter the scene after a background image is captured. These systems also use techniques such as face detection which require high resolution images. Instead of using background subtraction, fidget detection uses a combination of frame differencing and temporal histograms. Our technique does not require a background image and works well with low resolution images. In our system, audio and video are captured using four omni-directional microphones and a web camera respectively. Fast Bayesian acoustic localization on the captured audio gives an estimate of the speaker location. Fidget detection on the captured video is used to extract participant mug shots. The acoustic localization results are mapped onto the fidget detection results. Meeting participants are tracked by using a short-term time histogram approach. The system was successfully used to analyze meetings involving both stationary and moving participants. The results presented demonstrate the robustness of our system.

Dedication

I dedicate this work to my family.

Acknowledgments

I am thankful to my adviser, Dr. Stanley Birchfield for guiding me all along this endeavor. Thank you for your guidance, patience and encouragement, without which I would never have been able to complete this work.

I would also like to thank Dr. John Gowdy and Dr. Stephen Hubbard for being so accommodating. Thank you for gracing my committee with your presence.

Thanks to Miheer and Prashanth for helping me conduct the experiments.

Finally, I would like to thank my family for their love, patience and sacrifice.

TABLE OF CONTENTS

Table of Contents (Continued)

Page

List of Tables

List of Figures

List of Figures (Continued)

Figure Page

Chapter 1

Introduction

1.1 Importance of Meeting Capture and Analysis Systems

Almost every one of us has attended some kind of meeting in their lifetime. There are many kinds of meetings we attend in our day to day life: staff meetings, project reviews, group discussions, sales meetings, public presentations, class presentations and so on. We cannot overestimate the importance meetings play in our life. In North America business alone, about 17 million meetings are held daily [21]. The meetings and events industry is a $ 122.3 billion industry (Source: Convention Industry Council). Dallas-based Meeting Professionals International (MPI) [1], the largest association for the meetings profession, has more than 20,000 members in 68 chapters and clubs across the USA, Canada, Europe and other countries throughout the world.

It would be nice to have a system that could attend all our meetings, take notes and later generate reports. While this is not entirely possible, a meeting capture and analysis system aims to serve this goal. A meeting capture and analysis system can be useful for a variety of reasons:

1. Often due to schedule conflicts we are unable to attend important meetings. Also at times we do not want to take active part in a meeting and are only partially interested

in what happened during the course of the meeting. A meeting capture and analysis system can enable us to extract only relevant information from meetings we missed.

2. We often take notes during meetings. But not every one of us can do two things at a time. If we concentrate more on taking notes, we may not be able to take active part in the discussions. Note taking is an important but distracting task. A meeting capture and analysis system can generate automated notes for us.

3. Important decisions are very rarely made during the limited time period of a meeting. Most project managers like to review the proceedings of a meeting before they decide on the tasks to be performed. A captured and analyzed meeting can be later useful for providing additional context, details, and critical decision making.

1.2 Goals of a Meeting Capture and Analysis System

In order to be accepted by the common man, a meeting capture and analysis system must be well designed. The design must focus on both the technical and social aspects of meeting capture. To cater to a variety of users, a meeting capture and analysis system must meet the following goals:

1. The system must be non-obtrusive to the meeting. Participants of a meeting do not wish to be distracted. It is also important that the system be automatic. A manual system hampers the meeting dynamics.

2. The system must be able to capture a variety of multimedia information: audio data, video data, computer presentations, whiteboard drawings, personal notes, etc.

3. The usefulness of the system is primarily decided by its post-processing capabilities. The system should be able to extract relevant and important information from a recorded meeting: participant mug-shots, clustered audio data, indexed presentations, meeting information (date, time, number of attendees), etc.

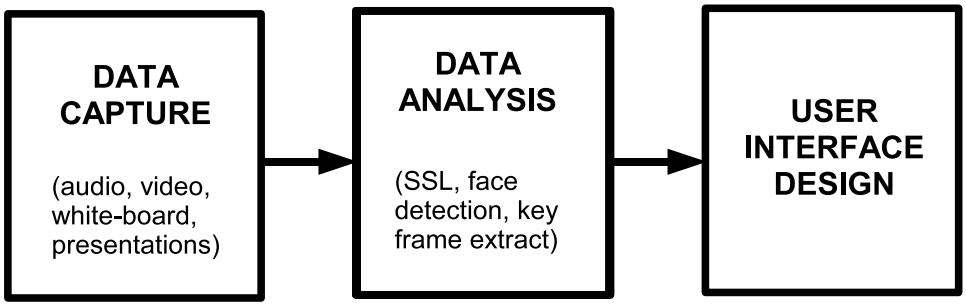


Figure 1.1: Sub-systems of a Meeting Capture and Analysis System (SSL = Sound Source Localization)

4. The system must have a user friendly interface.

5. The system must be reliable and affordable.

1.3 Related Work

Current research in the area of Meeting Capture and Analysis focuses on one or more of the following sub-systems (Figure 1.1): Data Capture, Data Analysis, and User Interface Design. There is a significant overlap between the Data Analysis and User Interface sub-systems. We will now take a look at each of these sub-systems as implemented in current research and commercial systems.

Data Capture: The capturing devices focus on capturing a variety of multimedia information: audio (speech), video (still images, video streams), whiteboard data (drawings, notes), computer presentations, etc. Distributed Meetings (DM) [8], a broadcasting and meeting recording system being developed at Microsoft, uses a 360-degree camera with integrated microphone array for audio-video capture. The 360-degree field of view is achieved by connecting five IEEE 1394 cameras in a pentagonal pattern. The camera system outputs a 3000 by 480 resolution panoramic image. The system also uses a 4 MP digital still camera to capture whiteboards and a 640 by 480 resolution overhead camera to provide a view of the entire meeting room. An 8-element microphone array with omni-directional mi-

crophones is used to capture 8 audio channels at 16-bit 44.1 KHz. Yong Rui et al. [17], at Microsoft Research, have developed a meeting analysis system using a high resolution omni-directional camera (1300 by 1030 = 1.3 Mega pixels). No audio recording is used. Instead, manual annotation is used to decide who is talking at each instant. CAMEO (Camera Assisted Meeting Event Observer) [18], a meeting capture and analysis system being developed at Carnegie Mellon University, consists of five firewire cameras mounted in a circle so as to provide an omni-directional view. To get a global description of the scene, CAMEO integrates the images coming from all the cameras into a single mosaic. Researchers at Ricoh Innovations have developed a portable meeting recorder [13] using an omni-directional camera, four microphones and a touch screen monitor.

Data Analysis: The captured data is processed in a variety of ways to extract important and relevant information. The DM system [8] uses generalized cross-correlation (GCC) for sound source localization (SSL). To improve audio quality, a delay and sum beamformer and automatic gain control (AGC) are used. The audio signal is band-filtered to remove non-speech frequencies, and a noise reduction filter removes stationary background noise. Person detection and tracking is implemented by integrating three modules: auto-initialization (combination of head and shoulder profiles, SSL and face detection), multi-cue tracking (based on Hidden Markov Models) and hierarchical verification (low-level and high-level). The whiteboard images are analyzed to detect key frames. The SSL and multi-person tracker information are combined to make intelligent decisions on what the user interface should show. The portable meeting recorder system [13] combines audio and video to extract participant mug-shots. SSL is used to find the approximate location, and then luminance variation and geometric feature analysis are used to identify skin pixels in the region of interest. They also use background/foreground extraction to identify the meeting location. Yong Rui et al. [17] use motion-detection and statistical skin-color tracking algorithms to track the meeting participants and extract mug shots. The CAMEO

system [18] uses a parts-based face detection algorithm for classification of image regions into “face” and “non-face” regions. Since this face detection algorithm is time consuming, background subtraction is used to obtain the regions of interest. Multimodal systems [21] using face ID, speaker ID, color appearance ID, and sound source directional ID have been developed to identify and track meeting participants.

User Interface Design: The user interfaces designed for meeting systems is dependent on the kind of data captured. On the other hand, the way this data is analyzed is dependent on what is to be shown in the user interface. Key features of the meeting browser client implemented in the DM system [8] are: timeline clusters for speaker segmentation; audio-video combination for intelligent decision making; time compression to remove pauses and to improve the playback speed; and time-stamping of whiteboard pen strokes. The main focus of the work done by Yong Rui et al. [17] is to study different meeting browsers. They study five different user interfaces: some show full-resolution video of all participants while others have only one main video window; some have overview windows while others do not; and some are user-controlled while others are computer-controlled. The Graphical User Interface(GUI) developed in CAMEO [18], allows the user to augment the captured video stream with information such as who was attending the meeting, and when they arrived and if they left early. The meeting browser developed by Ralph Gross et al. [10] lets users: create and customize dialogue, audio, and video summaries to the user’s particular needs; identify for each utterance the speaker properties (type, social relationships, and emotion). The portable meeting recorder system [13] has a meeting browser that augments the audio and video data with meta data which enables easy browsing of the meeting. Other features of their meeting browser are: automatic start and end time detection; viewable (TV show like) and searchable presentations; and best shot selection.

Other meeting systems have been developed that cater to a slightly different audience. “TeamSpace” [15] is an “Inter-Company Distributed Meeting System”, being developed

as a joint project between IBM Research, Boeing, and Georgia Institute of Technology. TeamSpace does not focus on capturing and analyzing audio video data. Instead, it focuses on coordinating meeting activities among participants at distant locations. A media enriched conference room for capturing meetings is being developed at FX Palo Alto Laboratory [6]. In their system, meeting participants manually capture video and presentation images. "Quindi Meeting Companion" [16], being developed by Quindi Corporation, is a personal software tool for capturing meetings. It primarily focuses on capturing meetings for personal use, just like note taking.

1.4 Thesis Outline

This thesis focuses on Data Capture and Data Analysis. The main contribution is the development of a novel approach to meeting video data analysis, called *fidget detection*. Fidget Detection is used to extract participant mug-shots. We have used off-the-shelf microphones and a web camera to build a simple system for capturing meetings. SSL is used to find the azimuth and elevation angles of the speaker. The thesis is organized as follows: Chapter 2 describes the approach taken to analyze audio and video data. Chapter 3 outlines the system architecture. Experimental results are presented in Chapter 4, and conclusions are drawn in Chapter 5.

Chapter 2

Meeting Analysis System - Approach

This chapter describes the approach taken to process audio and video.

2.1 Video Processing

All meeting analysis systems use some form of video processing. Video processing is used for a variety of purposes such as extracting participant mug-shots, extracting meeting room location, improving results of audio processing and so on. In this thesis, the principle technique used to process the video is fidget detection. Fidget detection combines frame differencing and temporal histograms. The participants may be present in the scene when the recording of the meeting starts. Fidget detection is used to extract participant mug shots during the entire course of the meeting.

2.1.1 Fidget Detection

The "Distributed Meetings" [8] system uses a vision-based multi-person tracker to track meeting participants and capture their mug shots. Their technique uses head-and-shoulder profiles, audio data and face detectors to initialize the tracker. The video processing requires high resolution expensive cameras. The video processing will fail when the image

resolution is poor. The “Portable Meeting Recorder” [13] system uses adaptive background modeling to extract the background. Background subtraction and sound localization are used to approximately localize meeting participants. Then mug shots are found by identifying the skin regions in these approximate locations. Background subtraction will fail if the participants are present in the scene before the system is started. Background modelling techniques such as averaging, mixture of Gaussians [9] are not useful in the context of meetings. These techniques work well when objects in the scene are moved. Though participants may change their seats in a meeting, there is no guarantee that this will happen. Thus backround modelling techniques cannot be applied to a meeting. Also the detection of skin pixels requires high resolution images (in their system video was captured at 640 x 480 resolution at 30 fps). This video processing will fail when the image resolution is poor.

Fidget Detection is a new technique to analyze meetings and extract participant information. Fidget detection is based on frame differencing and does not require any kind of background image. Hence participants are not required to enter the scene only after the system is switched on. Since we do not employ any kind of face detection or skin detection technique we do not require high resolution images. Instead temporal histograms are used to gather evidence about the location of participants. Fidget detection relies on the fact that even though a person is stationary for most of the time, there are instances in time when motion can be captured by taking the difference of two frames and integrating this motion over time will lead to the location of participants. In other words, a temporal histogram can be useful to determine participant locations.

Fidget detection can be broken down into the following steps: Motion Detection (Frame Difference, Thresholding and Morphological cleaning), Temporal Histograms, Height Estimation and Peak Detection. Figure 2.1 is a flow chart of the different stages in Fidget Detection.

We find the largest moving component at each time instant t. The difference of frames at time instant t and (t-1) gives a difference image. Many pixels undergo small changes due

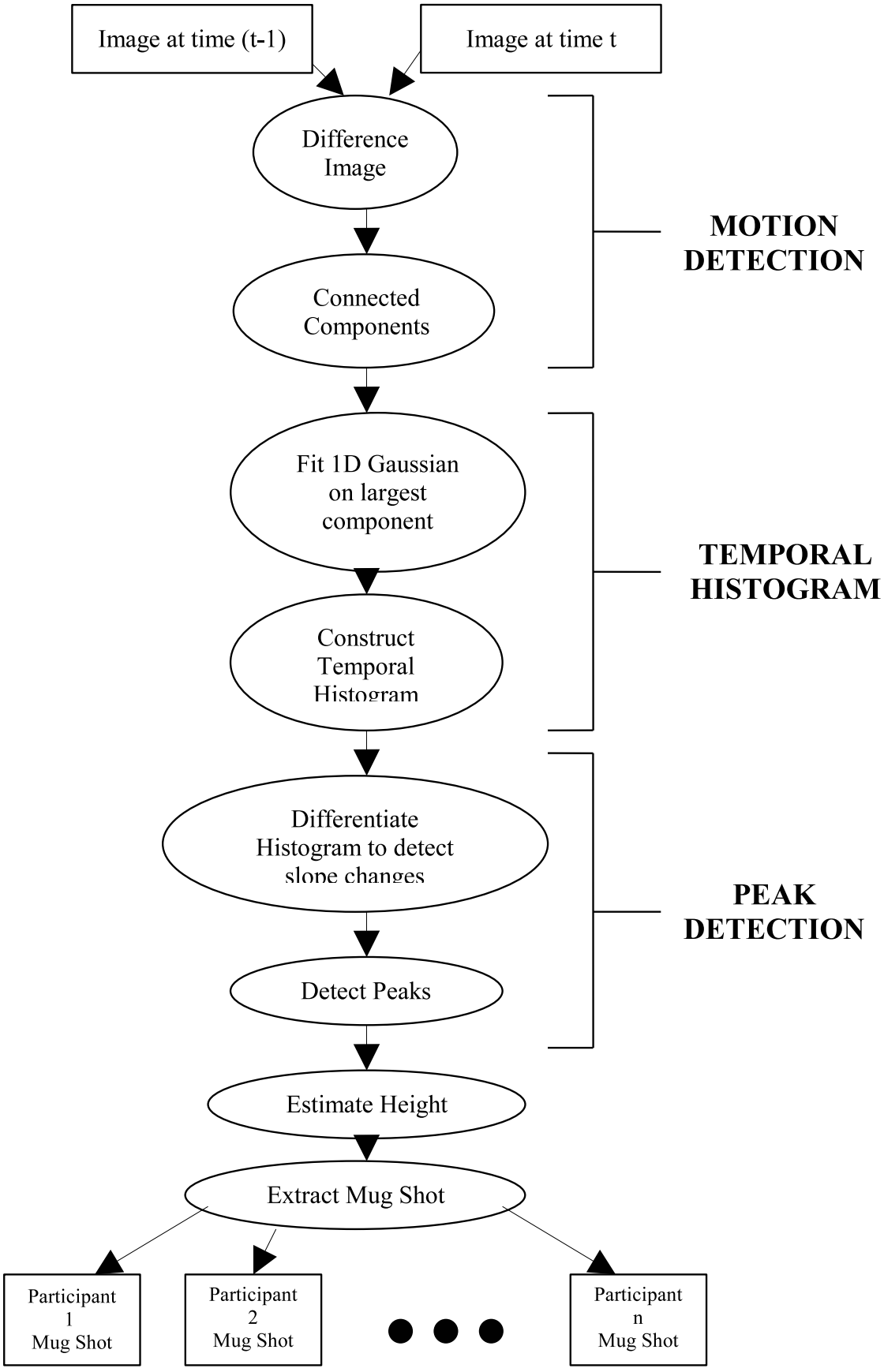


Figure 2.1: Fidget Detection Flowchart

to light variations. To eliminate detections due to light variations and only retain changed pixels due to participant motion, a binary image is created by applying a threshold to the difference image. This binary image has ones in regions where pixels undergo intensity changes due to meeting participant motion, and zeros otherwise. Some noise is also present in this binary image. We remove the noise by performing morphological opening (i.e. erosion followed by dilation) of the image using a 5-by-5 kernel of ones. In order to determine the largest region in the resulting image, connected components are found and the region with the largest area, and aspect ratio greater than a threshold is selected. Since we integrate the motion over time, all components other than the largest can be safely discarded. Even though this approach retains information about only one participant, eventually information about all participants will be gathered by integration.

A temporal histogram integrates the motion over time. A 1D Gaussian is fitted along the x coordinates of the largest region selected by the motion detection stage. The temporal histogram is constructed by continuously summing the Gaussian fitted at each instant of time. The histogram is smoothed by applying a low pass filter (i.e. by convolving with a [1 1 1] kernel).

In due course, peaks corresponding to each participant appear in the histogram. A simple algorithm is used to detect these peaks. The slope of the histogram at each spatial point is obtained by differentiation (i.e. by convolving with a [-1 0 1] kernel). All points where the slope changes sign from positive to negative are candidates for peaks in the histogram. The point with the largest magnitude in the histogram is declared as the location of the first peak. Assuming a fixed distance from the camera, all candidate points within a span (an input parameter to the peak detection routine) of the first detected peak are discarded. From the remaining candidate points, the point with the largest magnitude in the histogram is declared as the location of the next peak. This process is repeated until all the peaks are detected.

To extract mug shots we need the location of the face of each participant. The peaks in the temporal histogram correspond to the x location of the faces. To find the y location (i.e. height) of the face we maintain a history of the upper boundary of the largest region selected by the motion detection stage. The face of the participant is expected to be close to the top boundary. In order to account for noise, we select 90 percent of the top boundary as the height of the participant. The intersection of the histogram peaks and estimated heights correspond to the location of participant faces. Mug shots are extracted around these intersection points.

Figure 2.2 shows sample images from different stages of Fidget Detection. (a) and (b) are the frames captured at time instance t-1 and t respectively. The difference image (after thresholding) is shown in part (c). Part (d) shows the results of morphologically cleaning the image of part (c) and then finding connected components. Part (e) is the largest component selected. A Gaussian is fitted to this component. The temporal histogram at time t is the sum of Gaussians up to time t. Part (f) shows this histogram. The vertical lines correspond to the peaks and valleys in the histogram. There are false peaks in the histogram. Aassuming a fixed distance from the camera, all peaks within a span p of a detected peak are elminated. Hence, once a peak is detected all false peaks in its vicinty are eliminated. Part (g) shows the output of peak detection and height estimation. The horizontal lines correspond to the estimated height of each participant and the vertical lines correspond to the peaks detected in the histogram. The estimated height and histogram peaks are used to extract mug shots. These are shown in Part (h).

2.1.2 Fidget Detection: Tracking Moving Participants

The technique described in section 2.1.1 works very well if the participants do not change their position during the course of the meeting. When participants do move around, the histogram peaks in the original position of the participants leads to the detection of false peaks. To handle this scenario where participants cahnge their positions, we modified

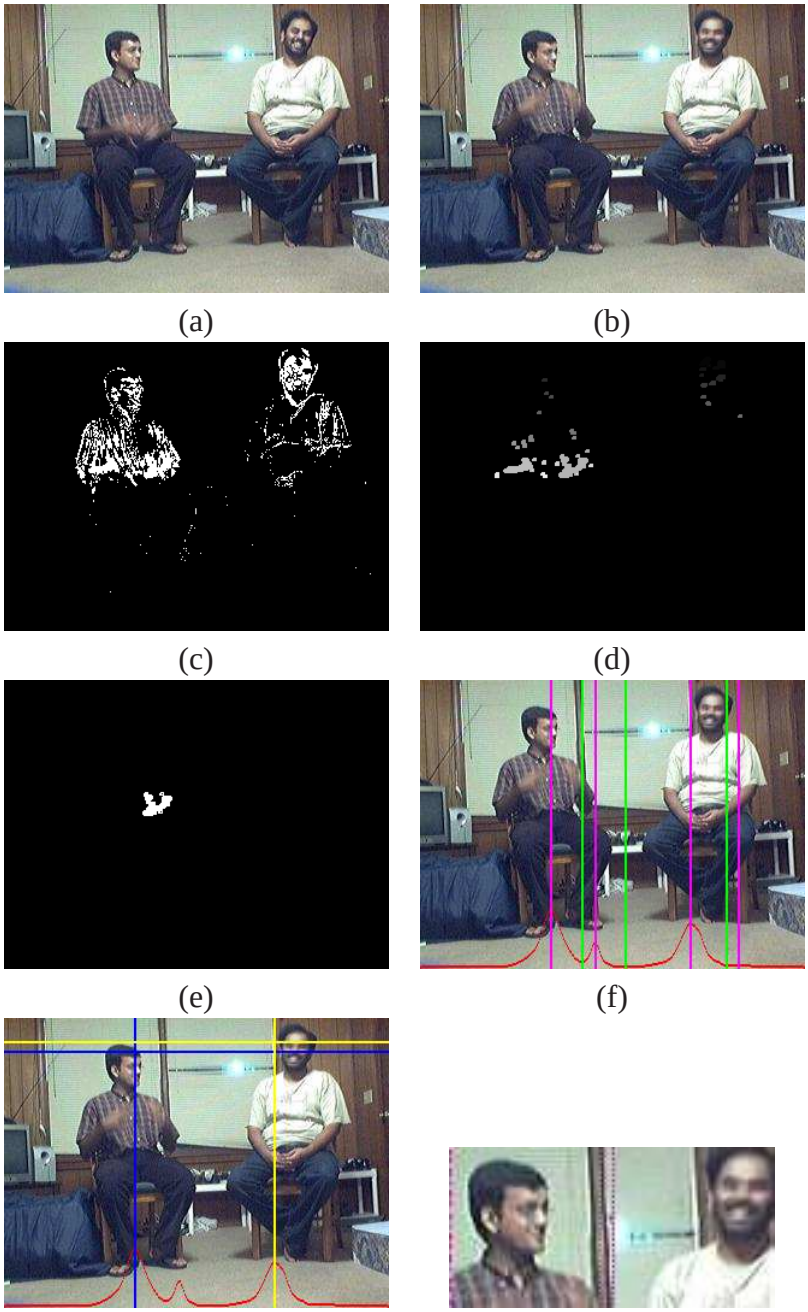


Figure 2.2: **Fidget Detection: Output at different stages** (a) Frame t-1 (b) Frame t (c) Difference Image (after thresholding) (d) Connected Components (after morphologically cleaning) (e) Largest component (f) Peaks and Valleys in the Histogram (g) Detected Peaks (h) Extracted Mug shots

the way temporal histograms are constructed. In our experiments it was observed that the temporal histogram built during the first few frames primarily decided the vicinity in which the peaks were detected. As the temporal histogram evolves over time the peaks get sharper. Based on this argument we used only the N_F most recent motion detections to construct the histogram. When a participant moves from location A to location B, frame differencing will detect no further activity at location A. Since the temporal histogram is constructed only from the most recent motion detections, the peaks in the histogram at location A will gradually fade out. At location B frame differencing will detect participant activity. The peaks in the histogram at location B gradually grows. In due course, the peak at location A is completely subdued and a peak is detected at location B.

2.2 Audio Processing

The four microphones are arranged in a square shaped compact array. The audio is captured at 16-bit 48 KHz sampling rate. The principal technique used to process the audio is Fast Bayesian Acoustic Localization. The output of acoustic localization is the azimuth (ϕ) and elevation (θ) angle of the speaker at each instant of time.

2.2.1 Fast Bayesian Acoustic Localization

Fast Bayesian Acoustic Localization [5] is a computationally efficient approach to the acoustic localization problem. The algorithm can be broken down into the following two steps:

1. Constructing the sampled hemisphere and determining correlation indices
2. Finding the probability of the source being present at each candidate location and selecting the candidate location with the highest probability.

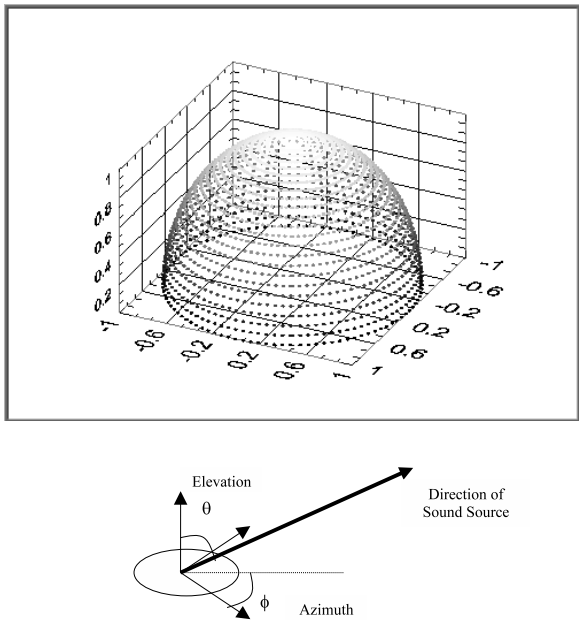


Figure 2.3: Sampled Hemisphere

The acoustic localization algorithm aims to find the probability of the source being present at locations on a hemisphere of unit radius around the compact array. Figure 2.3 shows a sampled hemisphere constructed around the microphone array. The hemisphere is divided into a fixed number of latitudes and longitudes. The acoustic localization algorithm determines the probability of the source being present at the intersection of each latitude and longitude. The points at the intersection of latitudes and longitudes are all candidate locations for the direction in which the sound source is present. The candidate location with the highest probability is selected as the most likely location of the sound source.

The probability of the source being present at location $\mathbf{q}$ and time t_0 is given by [5]:

$$\mathcal{L}_{corr}(\mathbf{q}) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \int_{t_0-W}^{t_0+W} x_i(t) x_j(t - \tau_{i,q} + \tau_{j,q})\, dt. \tag{2.1}$$

where N is the number of microphones, $2W$ is the window size, $x_i(t)$ and $x_j(t)$ are the signals arriving at microphones i and j, $\tau_{i,q}$ is the time sound takes to travel from location $\mathbf{q}$

to microphone i, and $\tau_{j,q}$ is the time sound takes to travel from location $\mathbf{q}$ to microphone j. The integral in equation 2.1 is the cross-correlation between the two signals x_i and x_j.

Since the microphone and candidate locations are fixed, the time sound takes to travel from a candidate location to the microphone location is fixed. Hence the time delay between the sound signals arriving at the two microphone locations from the same candidate location is also fixed. If we let $\mathbf{m}_i$ and $\mathbf{m}_j$ be the locations of microphones i and j, we can write

$$\tau_{i,q} = \frac{\| \mathbf{m}_i - \mathbf{q} \|}{c} \tag{2.2}$$

$$\tau_{j,q} = \frac{\| \mathbf{m}_j - \mathbf{q} \|}{c}, \tag{2.3}$$

where c is the speed of sound in the medium.

In discrete processing, when we cross-correlate the sound signals received at two microphones, the index in the correlation vector corresponding to the time delay $\tau_{i,j,\mathbf{q}} = \tau_{i,\mathbf{q}} - \tau_{j,\mathbf{q}}$ is given by

$$n_{i,j,\mathbf{q}} = \tau_{i,j,\mathbf{q}} * s \tag{2.4}$$

where s is the sampling rate.

Since c, m_i, m_j, $\mathbf{q}$ and s are all fixed and known offline, the correlation vector indices $n_{i,j,\mathbf{q}}$, for all the candidate locations on the sampled hemisphere and each microphone pair, can be determined offline.

Figure 2.4 shows the details of constructing a sampled hemisphere and determining the correlation index for the microphone pair 1 and 2. The hemisphere is sampled into number of latitudes by number of longitudes locations. For each location $\mathbf{q}$, equations 2.2 and 2.3 are used to find the time sound takes to travel to microphones 1 and 2. Equation 2.4 is used to find the correlation-vector index corresponding to location $\mathbf{q}$ and microphones 1 and 2. This process is repeated for each of the six microphone pairs: 1 and 2, 1 and 3, 1 and 4, 2 and 3, 2 and 4, 3 and 4.

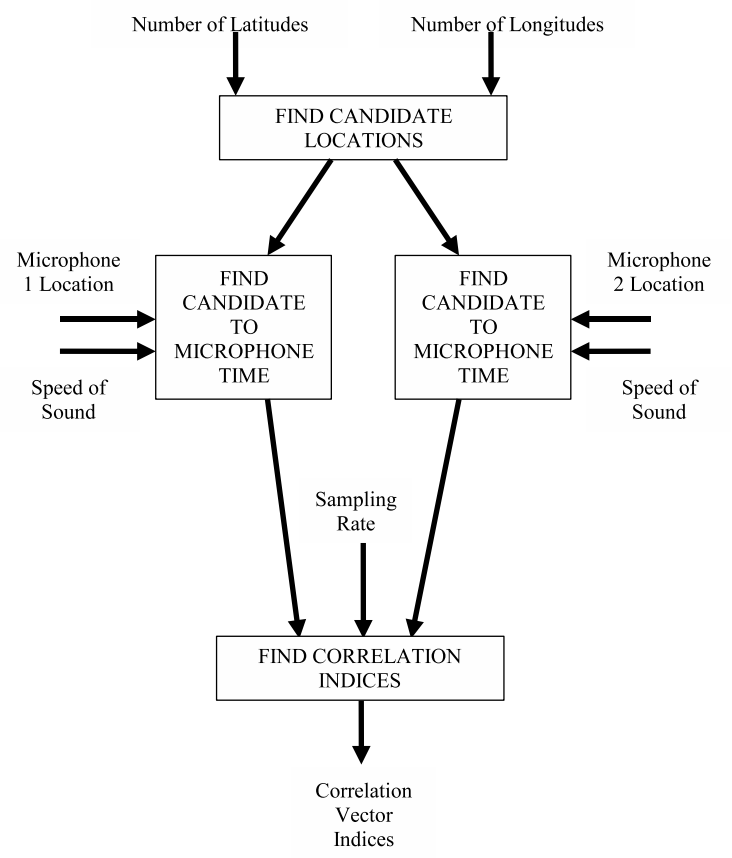


Figure 2.4: Determining Correlation Vector Index

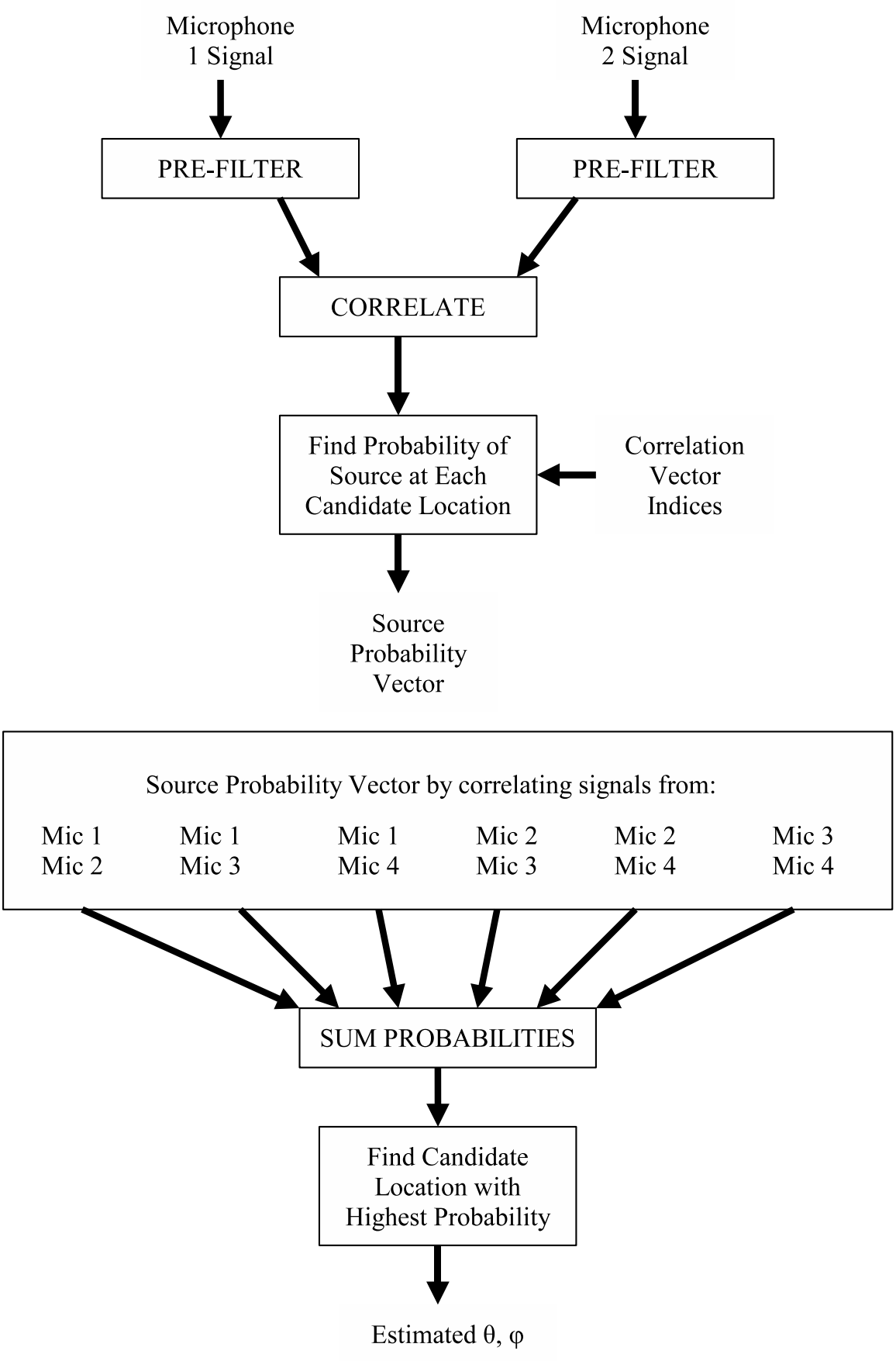


Figure 2.5: Estimating Azimuth and Elevation Angles

Figure 2.5 shows how the azimuth and elevation angles are estimated from the sound signals arriving at the four microphones. The sound signals are filtered using a PHAT filter [12, 14]. This pre-whitening filter improves results in the presence of reverberation. The filtered sound signals from two microphones are correlated. The magnitude of the resulting cross-correlation vector can be used as an estimate for the probability of different time delays between the sounds reaching the two microphones. The sound from each candidate location on the unit hemisphere will reach the microphones at a fixed time delay. The time delay corresponding to a particular candidate location is stored in the correlation vector indices (determined offline) (Figure 2.4). This time delay is used to index the correlation vector and find the probability of the source being present at that location on the sampled hemisphere.

A cross-correlation vector corresponding to each of the six microphone pairs is computed. From each correlation vector the probability of the source being present at each candidate location is obtained. The sum of the six probabilities corresponding to the six microphone pairs yields the total probability of the source being present at each candidate location.

The candidate location with the highest probability is declared as the most likely location of the speaker. The azimuth and elevation angle corresponding to the speaker location are obtained by simple geometry of the sampled hemisphere.

2.2.2 Mapping Audio Results

In order to relate the results of acoustic localization and fidget detection,the azimuth and elevation angles are mapped onto the image frame. Figure 2.6 shows the camera field of view along the horizontal and vertical planes. The camera is placed at the center of the microphone array. In figure 2.6, the camera field of view spans the horizontal plane for an azimuth angle of 45 degrees to 135 degrees. If the azimuth angle determined by the acoustic localization algorithm is within this range of 45 degrees to 135 degrees, the sound source

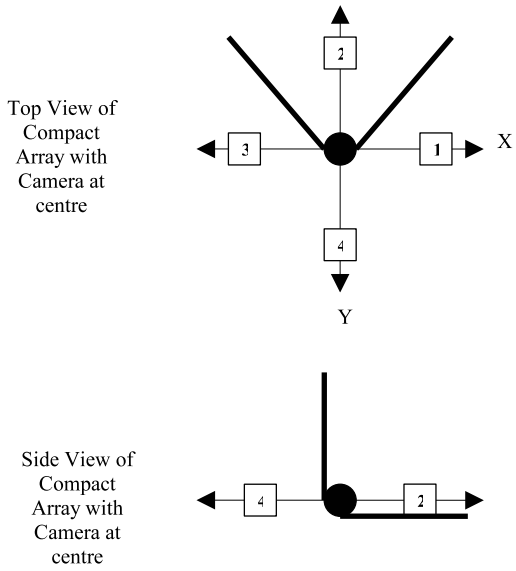


Figure 2.6: Camera Field Of View

is one of the meeting participants. All other azimuth angles are from a source outside the meeting and are discarded. The range of 45 degrees to 135 degrees is divided into slots corresponding to the number of columns in the image. The azimuth angle determined by the acoustic localization algorithm is mapped onto the corresponding column. Similarly, in figure 2.6, the camera field of view spans the vertical plane for an elevation angle of 0 degrees to 90 degrees. All other elevation angles are from a source outside the meeting and are discarded. The elevation angle determined by the acoustic localization algorithm is mapped onto the corresponding row in the image.

The pixel corresponding to the azimuth and elevation angle is the speaker location estimated using the acoustic localization algorithm.

Chapter 3

Meeting Analysis System - Implementation Details

The meeting analysis system hardware and software are discussed in this chapter.

3.1 Hardware Overview

A conceptual diagram of the Meeting Capture and Analysis system is shown in Figure 3.1. Figure 3.2 shows the actual system setup. The audio configuration consists of four omni-directional microphones arranged in a compact array. The microphones are arranged in a square of side 12 cm. Each microphone is a condenser element with wide range response and omni-directional pickup pattern. The simple microphone array provides 360 degrees acoustic capture without involving any complex construction. Figure 3.3 shows the wiring diagram used to power each microphone.

The microphone outputs are fed to a pre-amplifier stage. The pre-amplifier stage consists of two low-noise/high-gain pre-amplifiers designed specifically for use with microphones. Figure 3.4 shows a picture of the pre-amplififer used in our system. The outputs from microphones 1 and 2 are fed to pre-amplifier 1, and the outputs from microphones

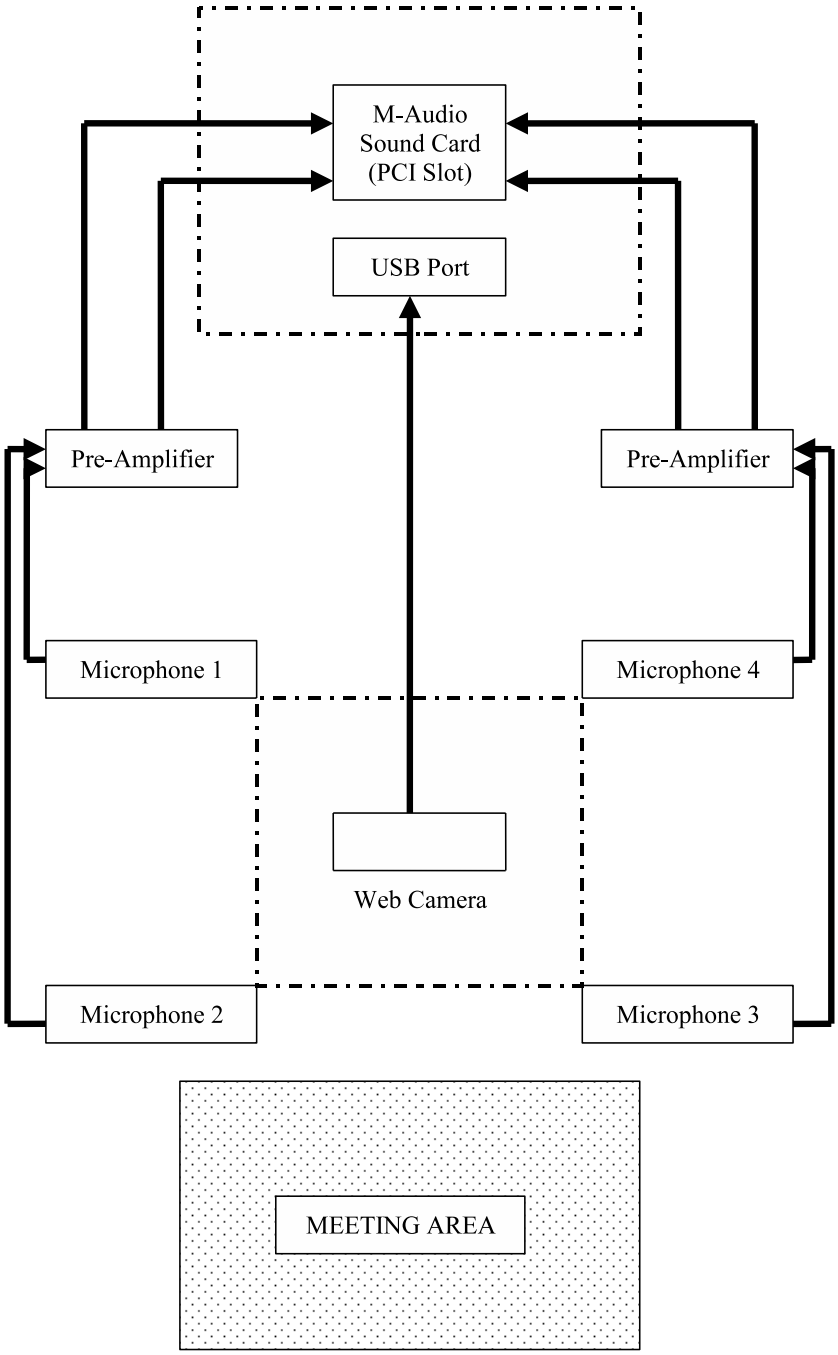


Figure 3.1: Meeting Capture and Analysis System Architecture

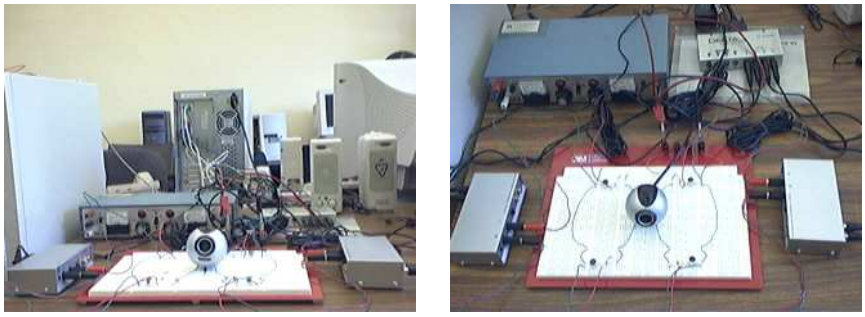
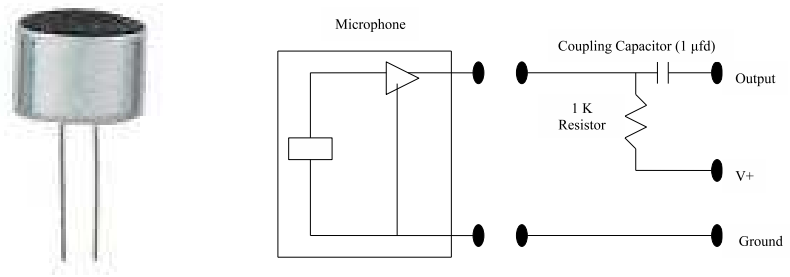


Figure 3.2: Meeting Capture and Analysis System Setup



(a) (b)

Figure 3.3: Microphone Condenser Element and Wiring Diagram

Figure 3.4: Microphone Pre-Amplifier

Figure 3.5: M-Audio Delta 44 Professional 4-in/4-out Audio Card

Figure 3.6: Logitech QuickCam Pro 4000 series USB Camera

3 and 4 are fed to pre-amplifier 2. The pre-amplifiers amplify the low-level microphone signals to line levels.

The amplified microphone signals are fed into a multiple-channel sound card. The Meeting Analysis system was built using the Delta 44 Digital Recording Interface by M Audio [2]. Figure 3.5 shows a picture of the Delta 44 sound card. The Delta 44 is a 4-input, 4-output digital recording interface capable of capturing high quality audio at data widths of 24 bit and sampling rates from 8 KHz to 96 KHz. In our setting, the Delta 44 card converts the analog line-signals to 16 bit 48 KHz digital audio signals. The sound card outputs two 16-bit signals. The signals from microphones 1 and 2 are interleaved to form signal 1. Similarly signal 2 is formed by interleaving signals from microphones 3 and 4. The two signals from Delta 44 are later processed in software to get the individual microphone signals.

Video is captured using a simple web camera. Our system uses a Logitech QuickCam Pro 4000 series USB camera. Figure 3.6 shows a picture of the USB camera. The web

camera outputs a 320 by 240 resolution video stream at 30 fps. The web camera is placed at the center of the compact microphone array. Its field of view is approximately 90 degrees in the horizontal and vertical planes. Because our web camera has no moving parts, it does not distract the meeting participants.

The audio and video streams are processed by a personal computer with an Intel(R) Pentium(R) 4 CPU operating at 2.80 GHz. The Delta 44 sound card is installed inside the Computer on a PCI slot.

3.2 Software Overview

The Personal Computer hosts the Meeting Analysis system software which performs all the tasks required to capture and process meetings. The software implemented using C++ makes use of DirectX [3] filters and Blepo Computer Vision Library [4].

The software architecture is based on object-oriented programming and can be broken down into the following five classes:

1. Meeting Analysis Class
2. Capture Direct Show Class
3. Capture M-Audio Class
4. Fidget Detector Class, and
5. Acoustic Localizer Class.

Figures 3.7 and 3.8 shows the communication details among different classes. The entire system is a multithreaded system with each class running in its own thread. The Meeting Analysis Class is the principal application class that creates instances of other classes. It is responsible for starting and stopping the different threads. It also coordinates the activity between all the classes.

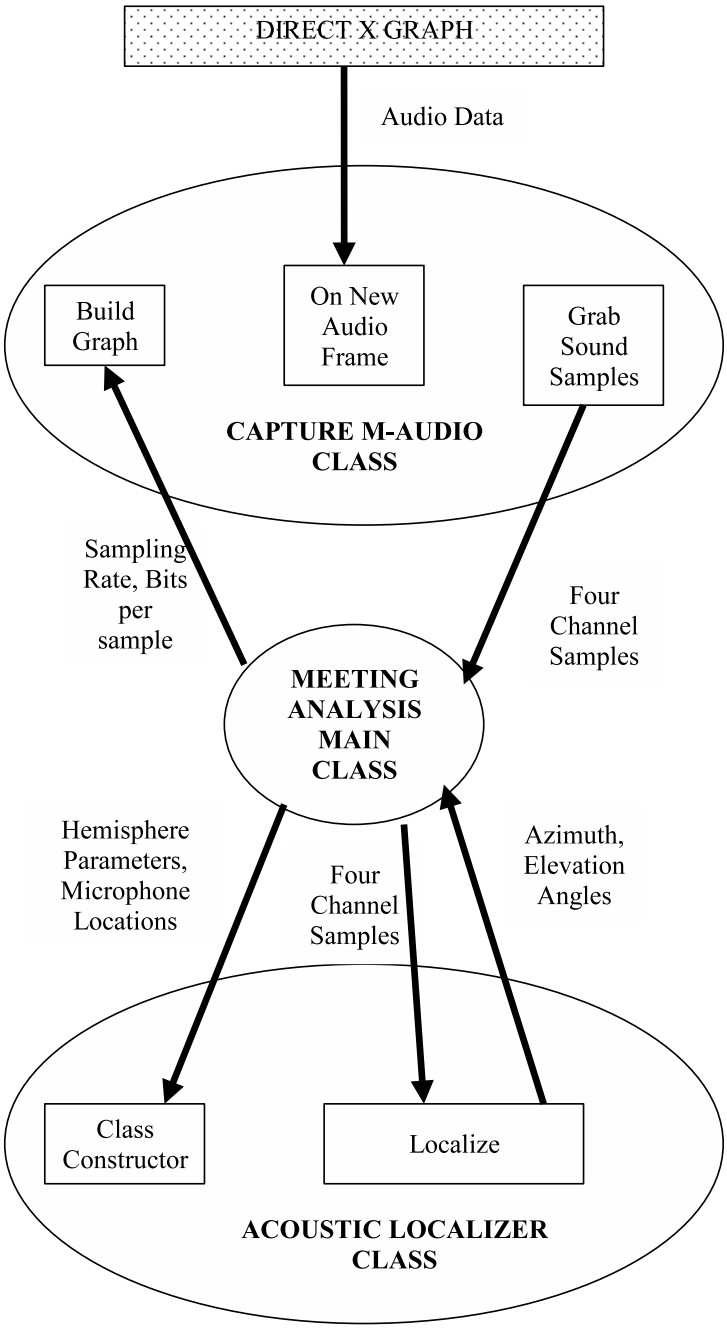


Figure 3.7: Software Diagram - Audio Capture and Processing

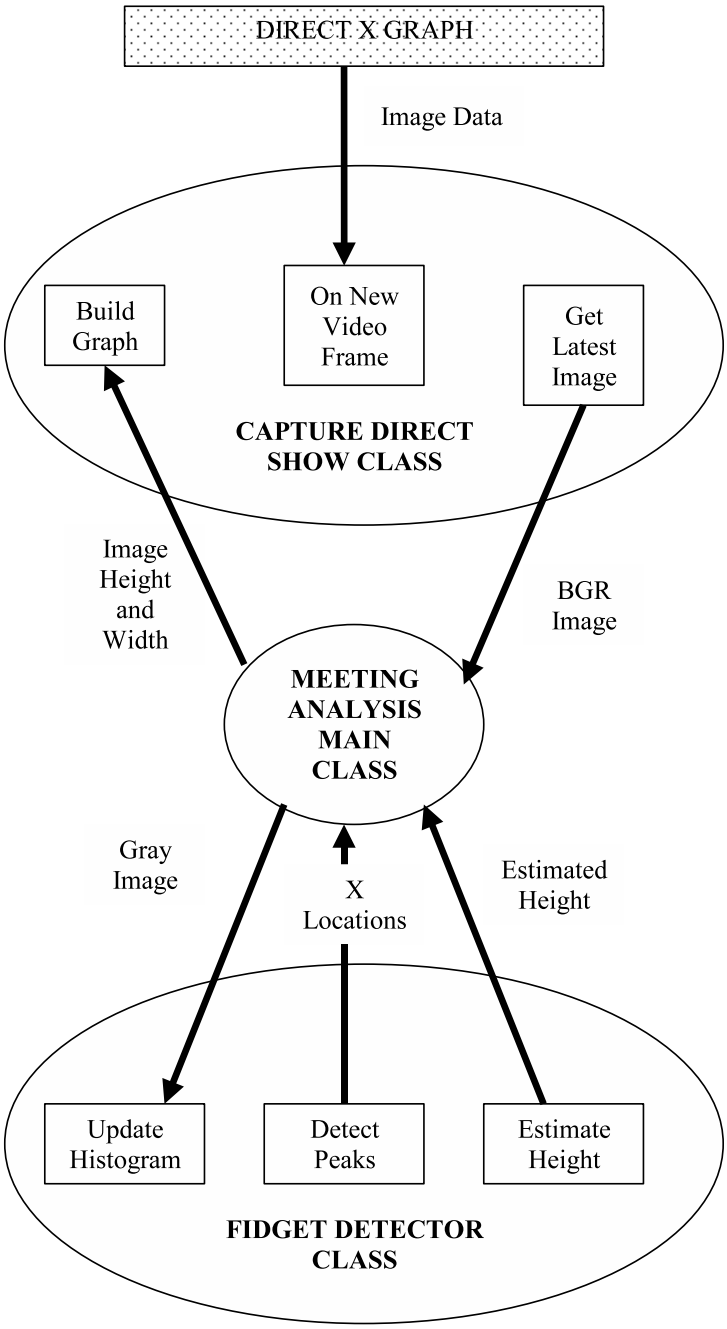


Figure 3.8: Software Diagram - Video Capture and Processing

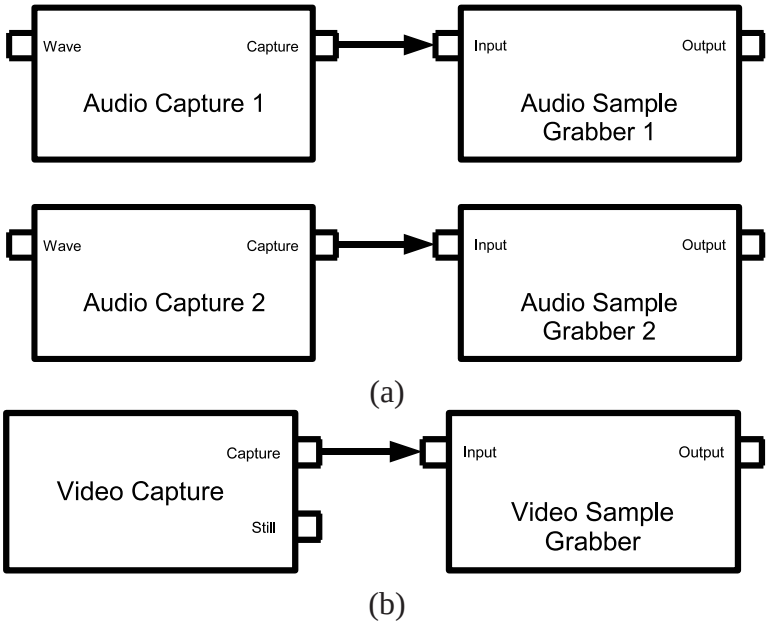


Figure 3.9: **DirectX Graphs:(a)** Audio **(b)** Video

The M-Audio sound capture card and Logitech QuickCam Pro 4000 series USB camera come with Microsoft DirectX compliant drivers. The Capture Direct Show (the class name is a misnomer) and Capture M-Audio classes are responsible for making calls to the video and audio drivers, respectively. Microsoft DirectX is an advanced suite of multimedia application programming interfaces (APIs) built into Microsoft Windows operating systems. DirectX provides a standard development platform for Windows-based PCs by enabling engineers to access specialized hardware features without having to write hardware-specific code. The various components of DirectX are in the form of COM-compliant objects. The Capture Direct Show and Capture M-Audio classes build DirectX graphs. These graphs are composed of different COM objects (also called filters) and define the way these objects are connected via their interfaces.

The Fidget Detector Class implements the Fidget Detection algorithm. It maintains a histogram and when asked returns the peak locations. The Acoustic Localizer Class implements the Fast Bayesian Acoustic Localization algorithm. It constructs a sampled hemisphere. When asked to perform localization it returns the azimuth and elevation angles.

When the Meeting Analysis system starts execution, the Meeting Analysis main class asks the Capture M-Audio class and Capture Direct Show class to build the DirectX graphs for audio and video, respectively. These graphs are shown in Figure 3.9. Each node in the graph is a DirectX filter. The Audio Capture and Video Capture filters correspond to the M-Audio DirectX and Logitech QuickCam Pro 4000 DirectX filters, respectively. The Audio Sample Grabber ia a DircetX COM object, part of the Capture M-Audio class. The Video Sample Grabber is a DircetX COM object, part of the Capture Direct Show class. The two audio graphs correspond to the two audio channels - one channel each for microphone pairs 1,2 and 3,4.

Once new video data is available, the Logitech QuickCam Pro 4000 DirectX filter notifies the Capture Direct Show Class and passes the image data. The Meeting Analysis class asks the Capture Direct Show class if new data is available. The Meeting Analysis class passes the image data to the Fidget Detector Class which maintains a time histogram as one of its member variables. The Fidget Detector class returns the peak locations in the histogram and the height estimate to the Meeting Analysis Class. The Meeting Analysis Class uses this information to extract mug shots of participants.

Once new audio data is available, the M-Audio DirectX filter notifies the Capture M-Audio Class and passes the audio data. The Meeting Analysis class asks the Capture M-Audio Class if new sound data has been captured by the sound card. The Meeting Analysis Class passes the sound data from all the four channels to the Acoustic Localizer Class. The Acoustic Localizer Class performs acoustic localization on the sound samples and returns the azimuth and elevation angles to the Meeting Analysis Class. The Meeting Analysis class is responsible for mapping the audio results onto the image frame.

Chapter 4

Experimental Results

The Fidget Detection algorithm was tested on two sequences captured at 30 frames per second using a single webcamera. Each sequence contains a meeting with two participants. The webcamera was placed on a flat table at a distance of approximately 2.3 meters from the meeting area. The sequences were digitized at 320 by 240 resolution. The first sequence was a meeting between two participants that lasted for approximately five and a half minutes. The sequence consists of 10,000 frames. During the course of this meeting participants change their seats. The second sequence was a meeting between two participants that lasted for approximately two minutes. The sequence consists of 4,000 frames. Through out the duration of this meeting participants maintain the same seating position. In both sequences, when the system is switched on the participants are already present in the scene. To test the effectiveness of various features of Fidget Detection, experiments were run on the two sequences.

In the first experiment, the Fidget Detection algorithm was tested on a sequence with moving participants. The sequence was first analyzed using short-term histograms. The sequence was then analyzed using long-term histograms. Figure 4.1 shows some results of using a short-term histogram. Figure 4.2 shows some results of using a long-term histogram. The histograms are superimposed on the image frames. The vertical lines cor-

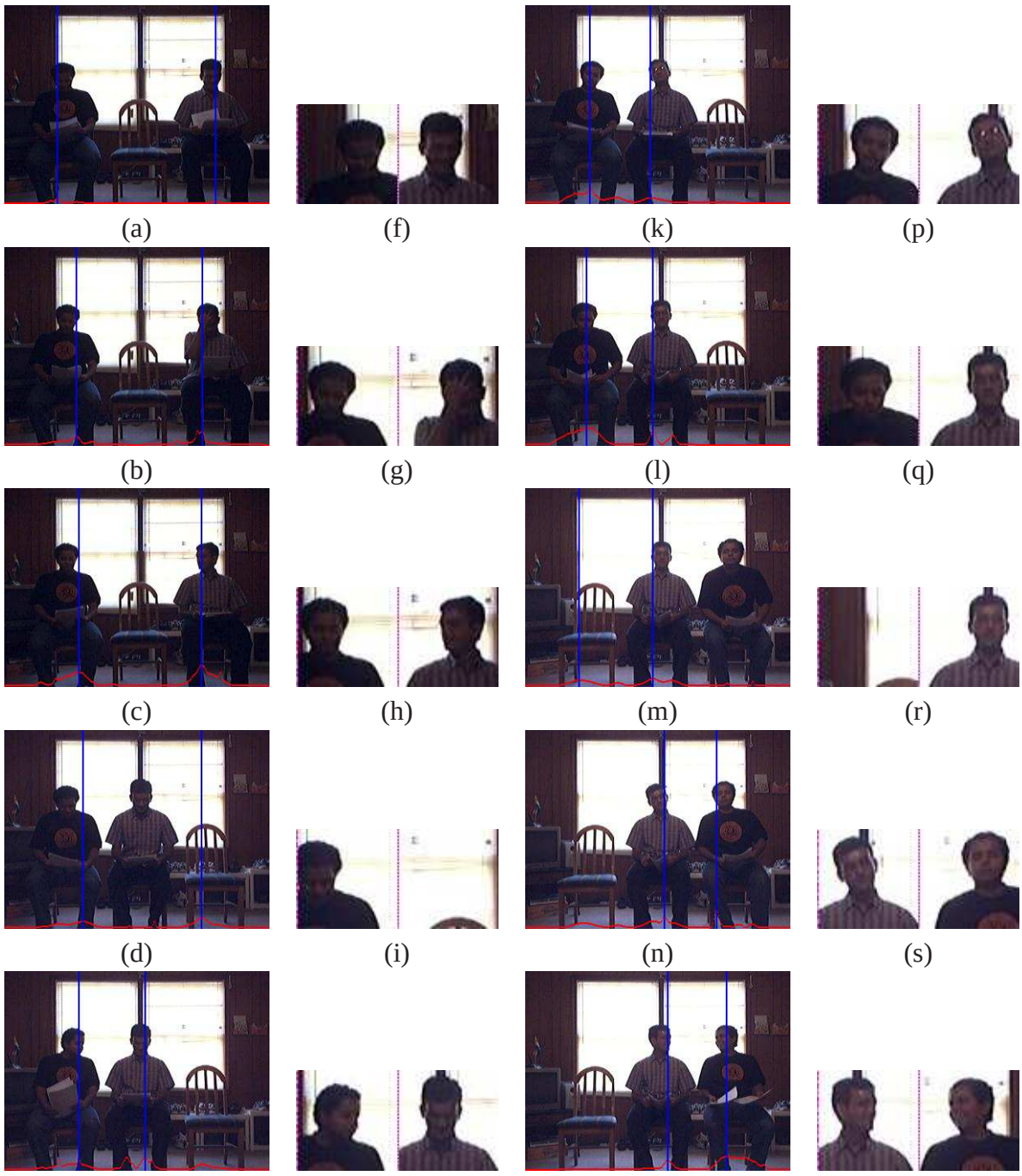


Figure 4.1: **Results using Short-Term Histograms on a Sequence with Moving Participants: (a) to (e)** Image Frames 192, 1895, 3600, 3837 and 4255 **(f) to (j)** Corresponding Mug-shots **(k) to (o)** Image Frames 5804, 7616, 7871, 7955 and 9328 **(p) to (t)** Corresponding Mug-shots

Figure 4.2: **Results using Long-Term Histograms on a Sequence with Moving Participants: (a) to (e)** Image Frames 192, 1895, 3600, 3837 and 4255 **(f) to (j)** Corresponding Mug-shots **(k) to (o)** Image Frames 5804, 7616, 7871, 7955 and 9328 **(p) to (t)** Corresponding Mug-shots

respond to the detected peaks in the histogram. Frame 3600 corresponds to the instance in the meeting when the participant on the right most seat is about to move to the seat in the center. Until this frame there is no observable difference in the mug-shots extracted using short-term and long-term histograms. Frame 3837 corresponds to the instance in the meeting when the participant has just settled down in his new seat. Both the short-term and long-term histograms are still locked on to the previous position of the participant. At frame 4255, the short-term histogram is able to successfully detect a peak in the histogram at the new location of the participant. The long-term histogram is still locked onto the previous position of the participant. In fact the long-term histogram continues to be locked onto the previous position of the participant until frame 7616. The short-term histogram is able to successfully extract mug-shots up to frame 7616. Frame 7616 also corresponds to the instance in the meeting when the participant on the left most seat is about to move to the right-most seat. Frame 7871 corresponds to the instance when the participant has settled in the right-most seat. The short-term histogram only detects the participant in the center seat. The long-term histogram only detects the participant in the right-most seat. This detection is due to the activity in the first 3600 frames of the sequence and is essentially a false detection. The short-term histogram quickly adjusts to the new seating position and is able to capture mug-shots by frame 7955. The long-term histogram completely loses track of the meeting and is not able to capture all the participant mug-shots till the end of the sequence (frame 9328).

In the second experiment, the Fidget Detection algorithm was tested on a sequence with stationary participants. The sequence was first analyzed using short term histograms. The sequence was then analyzed using long-term histograms. Figure 4.3 shows some results of using a short-term histogram. Figure 4.4 shows some results of using a long-term histogram. The histograms are superimposed on the image frames. The vertical lines correspond to the detected peaks in the histogram. The mug-shots extracted at frames 2851 and 2986 are slightly different for short-term and long-term histograms. While the long-term

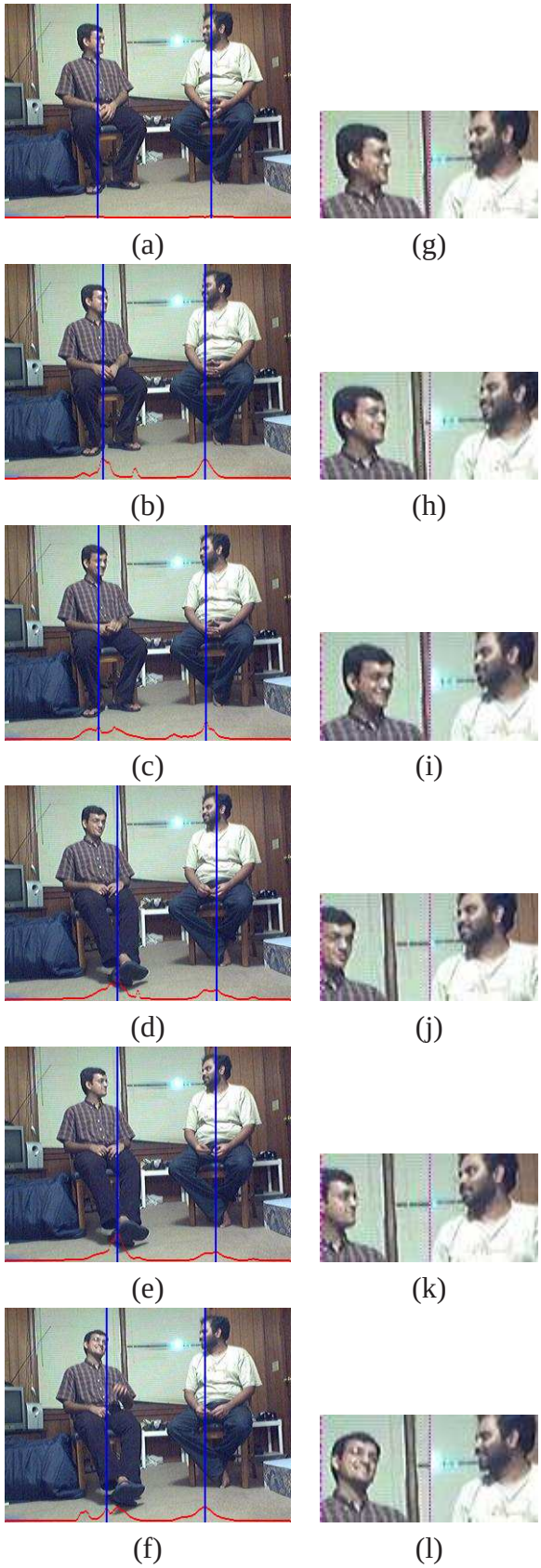


Figure 4.3: **Results using Short-Term Histograms on a Sequence with Stationary Participants: (a) to (f)** Image Frames 80, 574, 1425, 2851, 2986 and 3583 **(g) to (l)** Corresponding Mug-shots

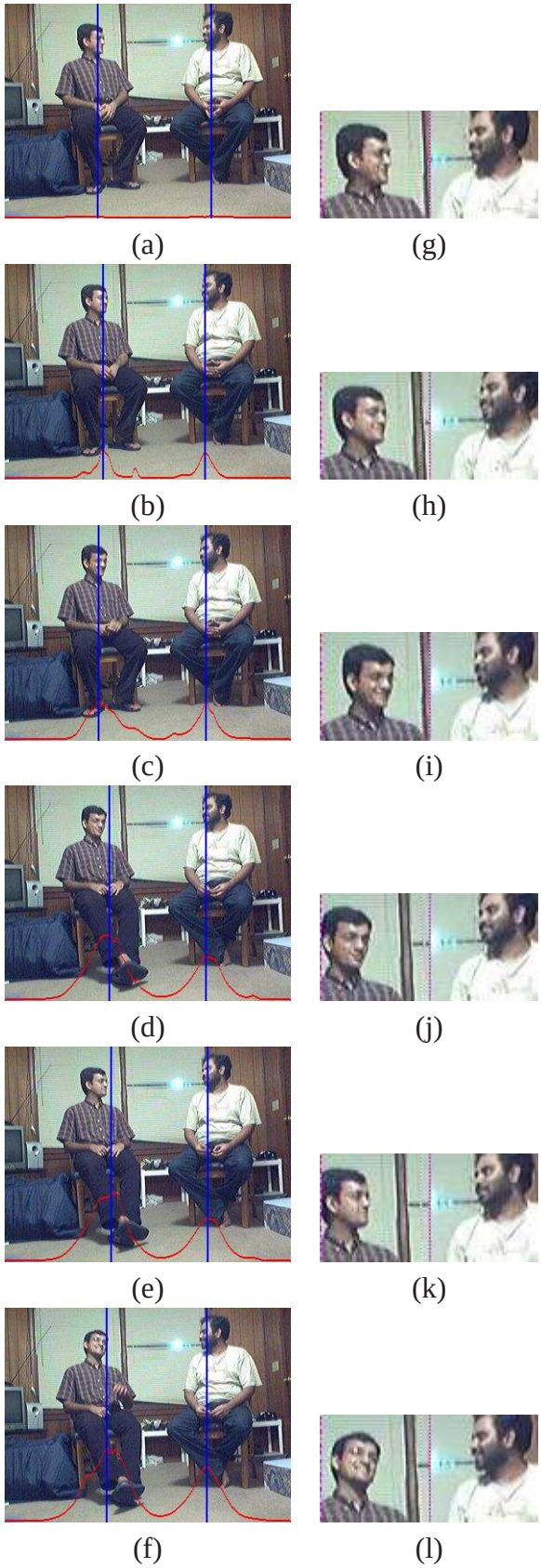


Figure 4.4: **Results using Long-Term Histograms on a Sequence with Stationary Participants: (a) to (f)** Image Frames 80, 574, 1425, 2851, 2986 and 3583 **(g) to (l)** Corresponding Mug-shots

histogram extracts almost complete mug-shots for both the participants, the short-term histogram extracts partially blocked mug-shots for the participant in the left seat. But for most practical cases this difference is negligible. The mug-shots extracted at all other frames are almost identical for short-term and long-term histograms.

The first two experiments demonstrate an important advantage of using short-term histograms. While short-term and long-term histograms generate almost identical results for meetings with stationary participants, short-term histograms are far superior to long-term histograms when participants change positions during the course of the meeting. In fact long term histograms are not even an option for meetings with moving participants. For meetings with stationary participants, long-term histograms generate slightly better results.

In the third experiment, the height estimated using Fidget Detection was compared with the ground truth height. The sequence with stationary participants used in the second experiment was also used in this experiment. Some results from the sequence are shown in Figure 4.5. Long-term histograms were used in this experiment. The first column shows the image frames with the histograms superimposed. The vertical lines correspond to the detected peaks in the histogram. The second and third columns correspond to the extracted mug-shots and the ground truth mug-shots respectively. For almost all the frames the height estimated using fidget detection is within five percent of the ground truth height. The estimated height for the left participant is incorrect only at frame 2963. The estimated height for the right participant is incorrect at frames 63, 2396 and 2963. The incorrect height is due to more motion in the region around the hands.

In the final experiment, Fidget Detection and Fast Bayesian Acoustic Localization were tested on an audio-video recording (presentation). The recording equipment consisted of a compact microphone array and a webcamera. The webcamera was placed at the center of the microphone array. The microphones and the webcamera were placed on a flat table at a distance of approximately 1.3 meters from the meeting area. The camera field of view spans the horizontal plane from 90-degrees to 180-degrees and the vertical plane from

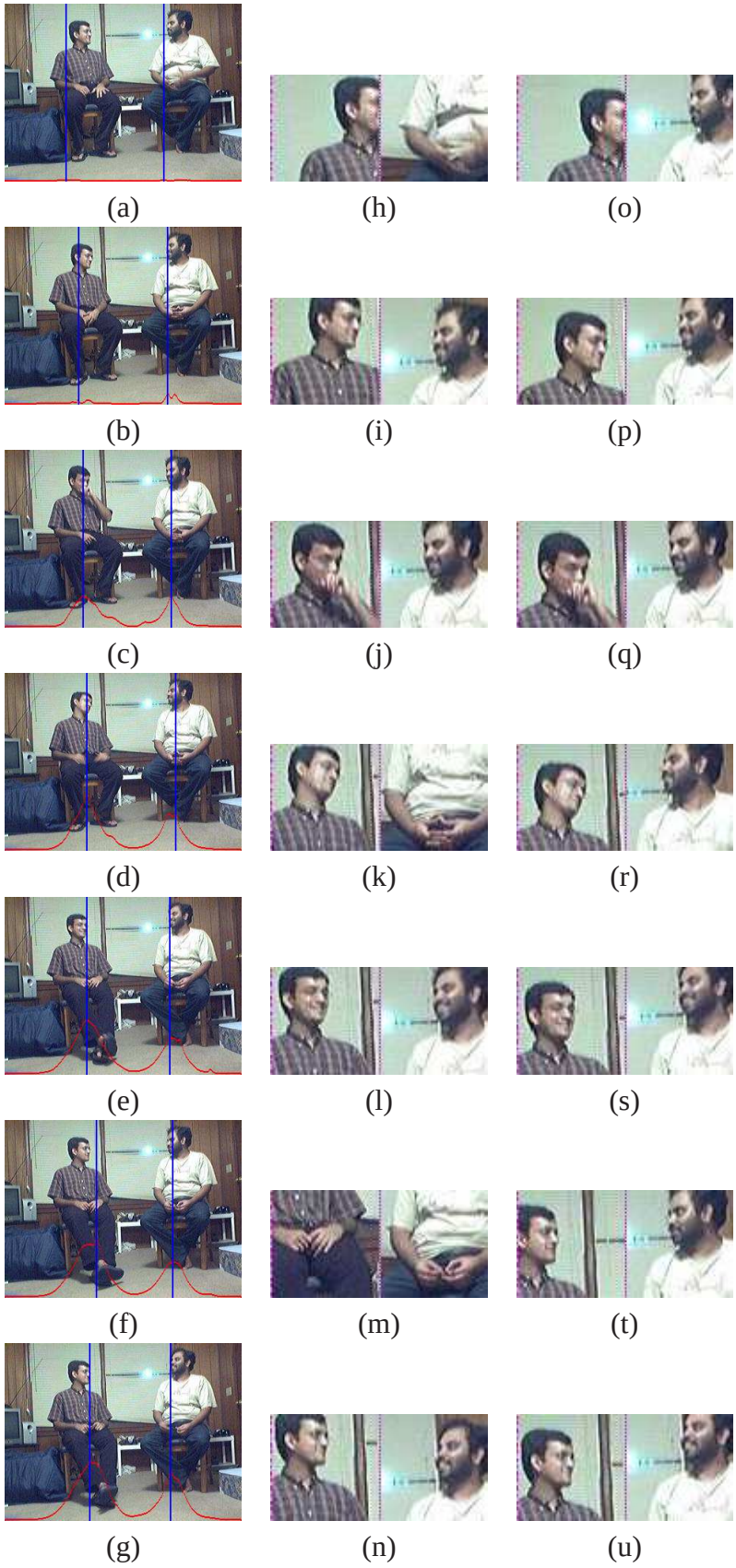


Figure 4.5: **Results of Height Estimation: (a) to (g)** Image Frames 63, 217, 1440, 2396, 2821, 2963 and 3333 **(h) to (n)** Mug-shots using height estimation **(o) to (u)** Mug-shots at a fixed height (ground thruth)

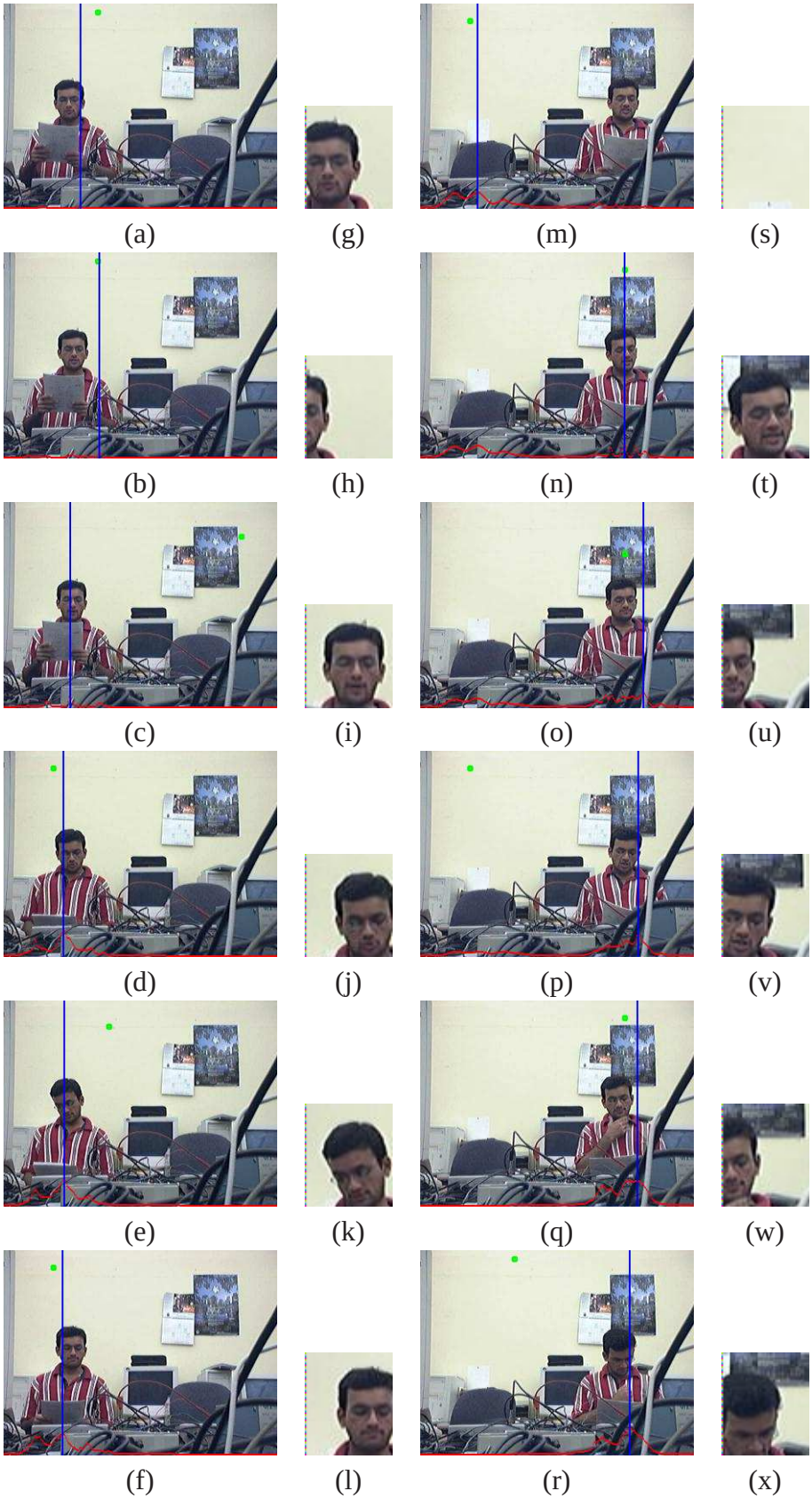
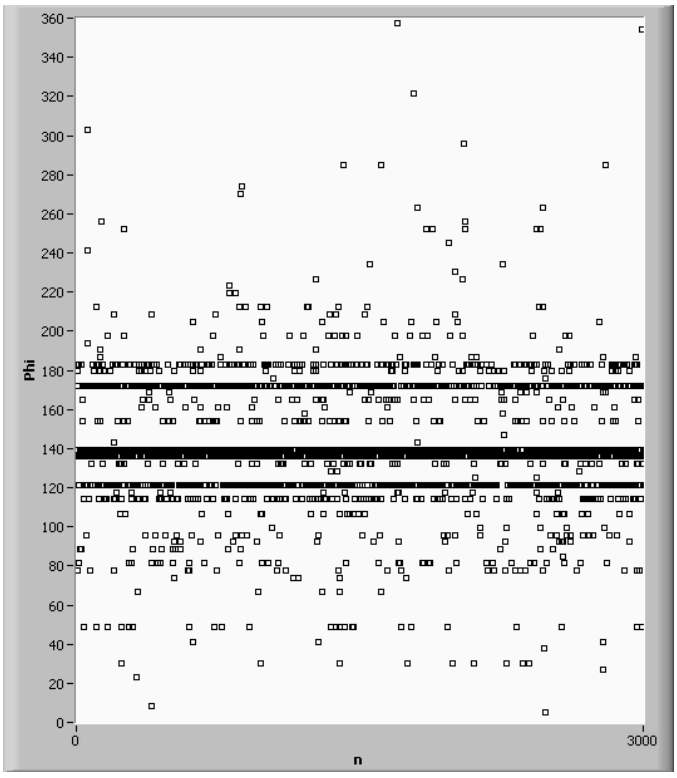
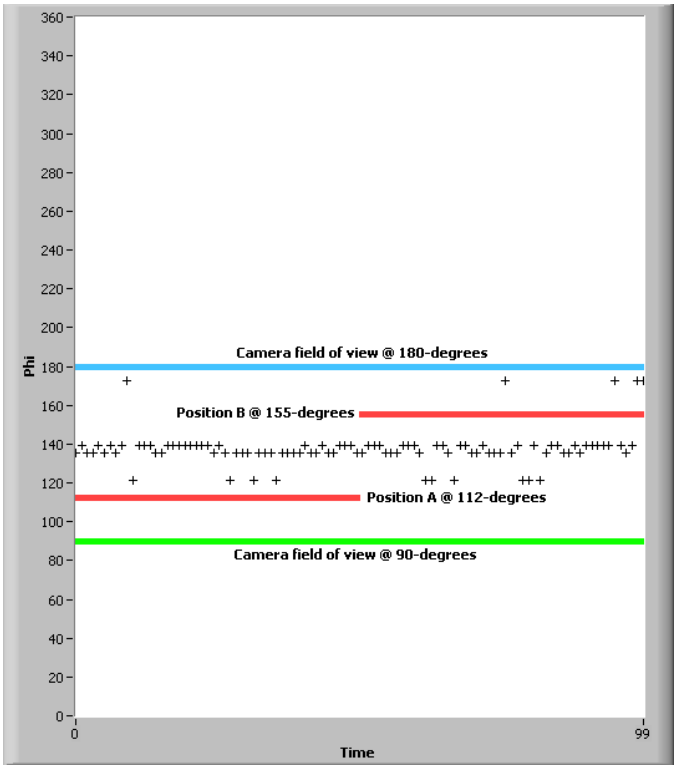


Figure 4.6: **Results of Acoustic Localization and Fidget Detection: (a) to (f)** Image Frames 11, 21, 139, 812, 1116 and 1477 **(g) to (l)** Corresponding Mug-shots **(m) to (r)** Image Frames 1613, 1813, 2009, 2217, 2648 and 2706 **(s) to (x)** Corresponding Mug-shots



(a) Actual Results



(b) Results after grouping 30 samples and slecting the highest frequency value

Figure 4.7: Acoustic Localization Results

Frame Number	Azimuth (ϕ)	Elevation (θ)
11	121	4
21	121	4
139	168	15
812	106	8
1116	125	11
1477	106	8
1613	106	8
1813	157	8
2009	157	23
2217	106	8
2648	157	8
2706	121	4

Table 4.1: Acoustic Localization Results (All angles are in degrees)

approximately 0-degrees to 90-degrees. The presentation consists of a single participant. The participant changes seats during the course of the meeting. Short-term histograms were used in this experiment. Fidget Detection was used to extract mug-shots and Fast Bayesian Acoustic Localization was used to calculate the azimuth (ϕ) and elevation (θ) angle of the sound source. The acoustic localization results are mapped on to the image frame. Some results from the sequence are shown in Figure 4.6. The histogram is superimposed on the image frames. The vertical lines correspond to the detected peaks in the histogram. The blobs correspond to the acoustic localization results. Table 4 shows the azimuth and elevation angles at different frames. Frame 1477 corresponds to the instance when the participant is about to change his seat. Frame 1613 corresponds to the instance when the participant had just settled down in his new seat. Figure 4.7 shows the azimuth angles plotted against time. Part (b) was obtained from Part (a) by forming groups of thirty samples, constructing a histogram with bins in the range of 0-degrees to 360-degrees and selecting the highest frequency bin. The acoustic localization algorithm is not able to accurately track the participant. There are many reasons for this: the audio was not captured in a noiseless and reverberation-free environment; the microphone locations, camera field of view, and participant ground truth locations are not accurate. Inspite of these experimental

limitations, the acoustic localization algorithm is successful in finding the correct quadrant (90-degrees to 180-degrees) in which the meeting is conducted. The acoustic localization work done in this thesis is in the early stages. The algorithm needs to be tested on more sequences. The results of acoustic localization can be used to improve the performance of the Fidget Detector and vice versa. Other reserachers have combined audio and video for a variety of applications [20, 11, 7].

As a final test, the Viola-Jones face detection algorithm [19] was run on all the three sequences. The code implemented in Blepo [4] was used for this test. The face detector algorithm was successful on the third sequence containing a single participant. But the algorithm failed completely on the first two sequences. As the first two sequences were captured with the camera at a distance of 2.3 meters, their image resolution is lower than the third sequence, which was captured with the camera at a distance of 1.3 meters. Thus as the image resolution is lowered the face detection algorithm fails. This demonstrates that meeting analysis systems that make use of face detection would fail on sequences captured at a lower resolution.

Chapter 5

Conclusion

The video processing in most meeting analysis systems is based on background subtraction and assumes that a clean background image will always be available. This need for a clean background image requires meeting participants to enter the scene only after the system is switched on. This thesis presents a new approach to meeting analysis which is free of such requirements. The main contribution is the development of a novel approach to video analysis. This approach, which makes use of frame differencing and temporal histograms, relies on how people fidget and thus we call it *fidget detection*. Fidget detection was used to extract participant mug shots. A system to capture meetings was built using simple off the shelf microphones and web cameras. Fast Bayesian acoustic localization was used to process the audio and find the direction of the sound source. A simple technique based on the geometry of the system was used to map the audio results onto the image frame. The meeting analysis system presented in this work is able to track meeting participants as they change their positions during the course of the meeting. The system was successfully tested on a variety of meeting recordings. Most meeting analysis systems would fail on the low resolution images captured using a web camera. Our system was able to successfully extract meeting information from these low resolution images.

The algorithms presented in this thesis were an initial attempt at analyzing meetings. Some aspects of our algorithm need further analysis and enhancements. The system was tested on meetings captured using a single web camera. The field of view of a single camera is limited. A more realistic meeting analysis system would make use of an omni-directional camera system. Such systems can be realized in more than one way - a single camera with omni-directional capabilities; multiple cameras can be arranged to capture a 360 degrees view. The 360 degrees panoramic images captured by these cameras enable the meetings to be conducted in a more realistic manner. Our system needs to be tested on meetings captured using omni-directional systems. In our initial investigations we used a simple approach to track participants as they changed their positions during the course of the meeting. More sophisticated techniques can be used to track participants. Also better techniques can be used to map the audio results on to the image frame. The results of audio processing can be used to enhance the results of video processing and vice-versa. An important aim of meeting analysis systems is to generate indexed meetings which can be easily browsed at a later time. Such post processing capabilities are an important area for further development in our system.

APPENDICES

Appendix A

Analysis of Acoustic Localization implementation

As discussed in Chapter 4, the acoustic localization implementation was not able to accurately track a participant when he changed his seats. In this section we analyze the implementation using a divide and conquer approach. The implementation is tested on a new set of data.

The Fast Bayesian acoustic localization implementation can be divided into the following sub-routines:

1. Find candidate locations on the hemisphere.
2. Find the correlation vector indices for each microphone pair.
3. Cross-correlate the microphone signals.
4. Combine the results of cross-correlation for all the six microphone pairs.

Figure A.1 was obtained by plotting the candidate locations. This figure demonstrates that the candidate locations form a sampled hemisphere.

Figure A.2 plots the correlation vector indices corresponding to the pair of microphones 1 and 3. Microphone 1 is at 0-degree azimuth and microphone 3 is at 180-degree azimuth.

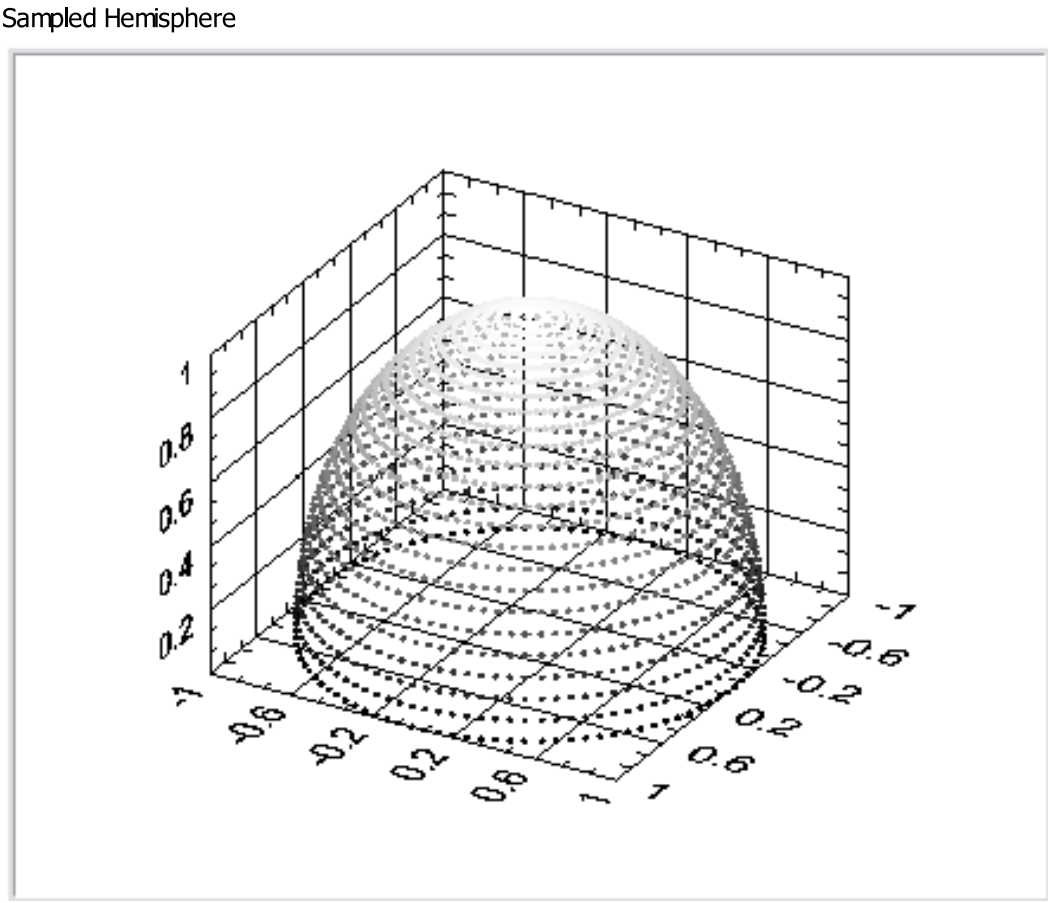


Figure A.1: Sampled Hemisphere

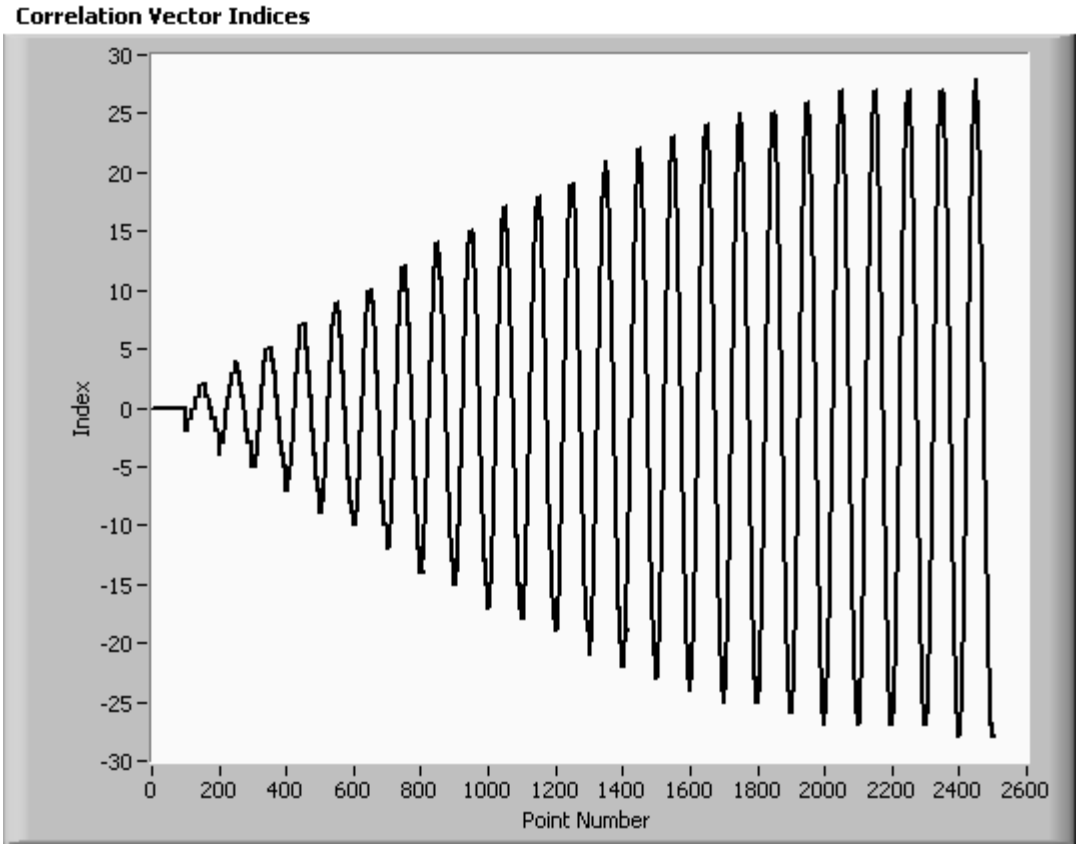


Figure A.2: Correlation indices for microphone pair 1 and 3.

The candidate locations are plotted beginning with the circle of points at the top and moving down towards the bottom. The maximum correlation vector index from this plot is 28. The distance between two adjacent microphones is 0.14 meters. Since microphones 1 and 3 are diagonally opposite, the distance between them is $(0.14) * \sqrt{(2)}$. For sampling rate 48000 and sound speed 345 m/s, the calculated maximum correlation vector index is 27.5. Thus the maximum correlation vector index from the plot matches the actual value.

New data was captured by using a white noise source. The source was initially placed at an approximate azimuth angle of 45-degrees and then at an approximate azimuth angle of 225-degrees. The Fast Bayesian acoustic localization implementation was run on the data captured by the four microphones. Figures A.3 and A.4 show the results. Figure A.3 plots the acoustic localization results as determined every 33 ms. Figure A.4 plots the results after grouping thirty points, constructing a histogram with bins 5-degrees wide and selecting the bin with the largest amplitude. These results demonstrate that the acoustic localization algorithm was able to successfully determine the location of the sound source.

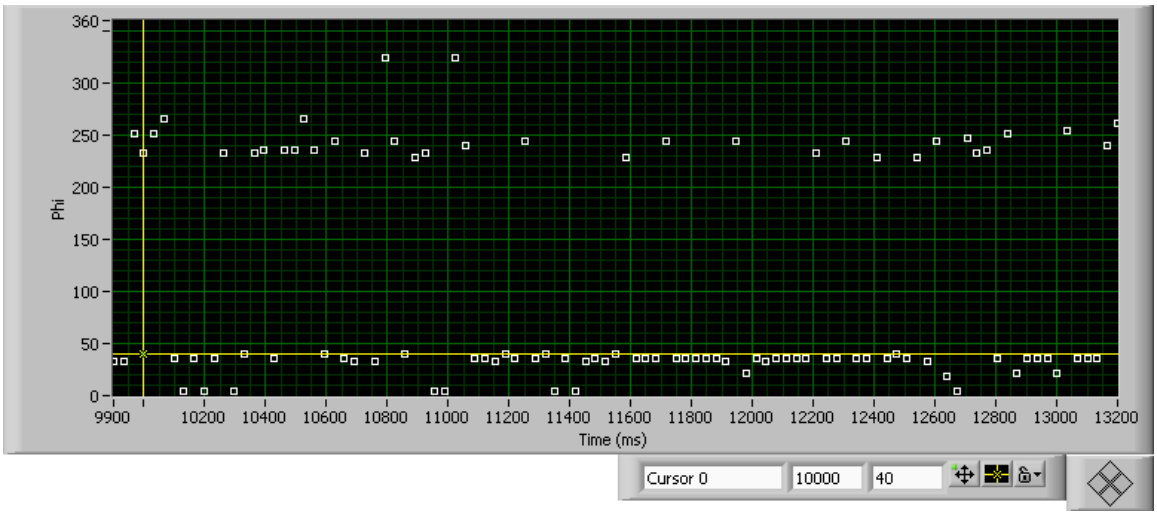
Figure A.5 shows the result of cross-correlating the signals received at microphones 1 and 3, using three different methods:

1. Direct cross-correlation as implemented in the acoustic localization implementation.

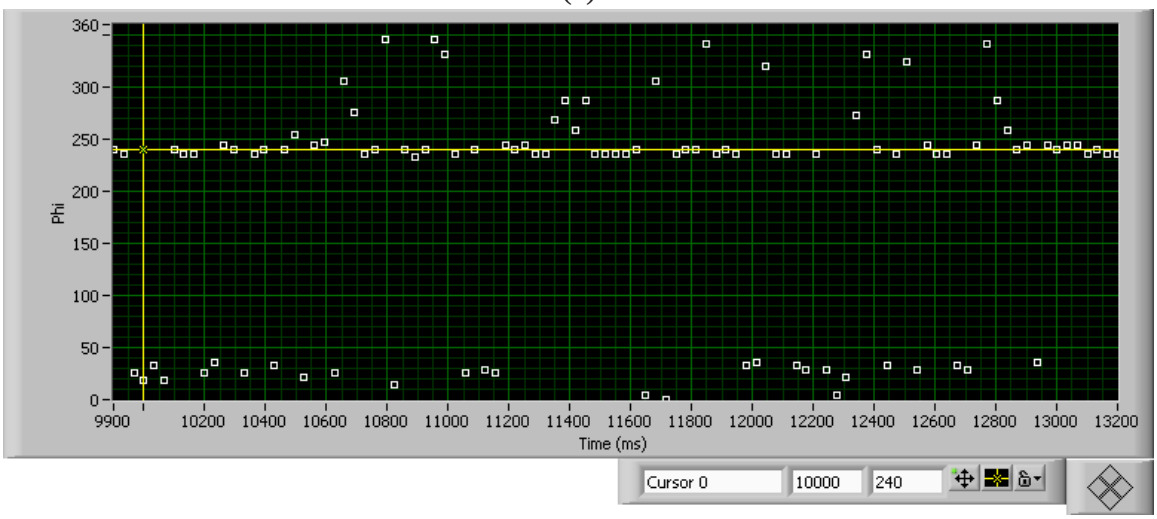
2. Cross-correlation using FFT as implemented in the acoustic localization implementation.

3. By cross-correlating the signals using a built-in LabVIEW function.

There is no visible difference between the results of the three techniques.

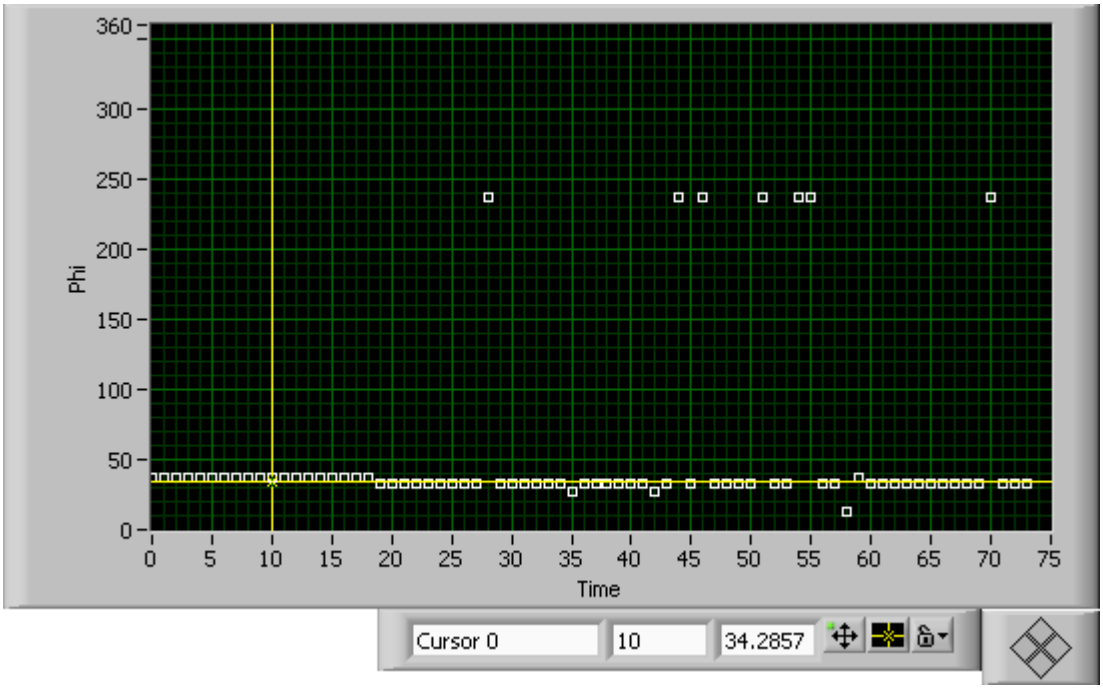


(a)

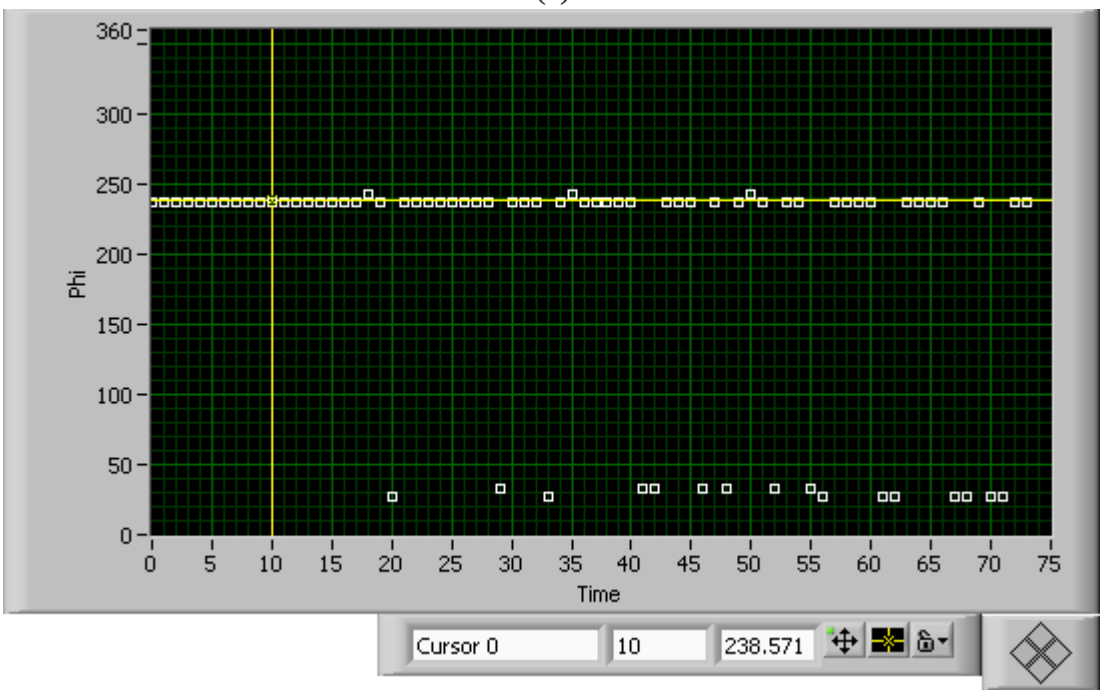


(b)

Figure A.3: **Acoustic Localization Results before Filtering** (a)Source at 45 degrees (b) Source at 225 degrees

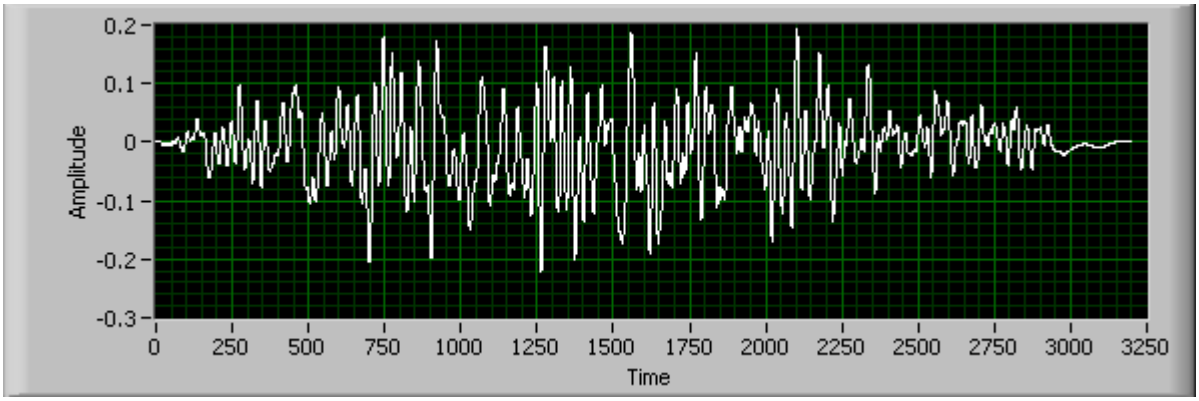


(a)

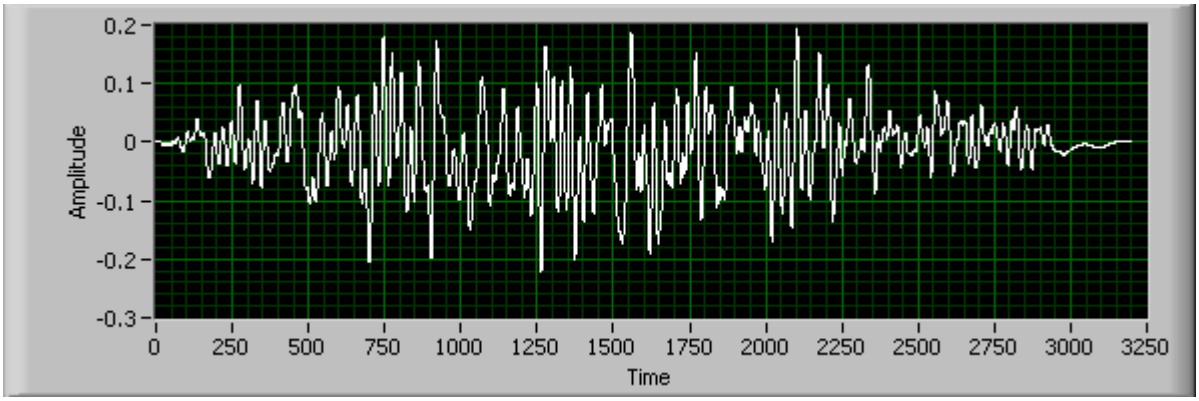


(b)

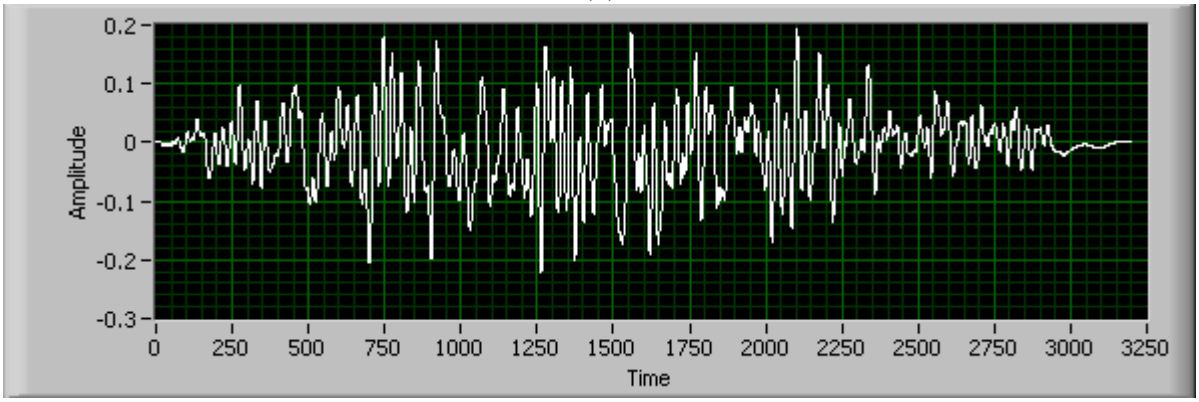
Figure A.4: **Acoustic Localization Results after Filtering** (a)Source at 45 degrees (b) Source at 225 degrees



(a)



(b)



(c)

Figure A.5: **Cross-Correlation** (a)Direct (b)Using FFT (c)Using built-in LabVIEW function

Bibliography

[1] Meeting Professionals International. http://www.mpiweb.org.

[2] M-Audio. http://www.m-audio.com.

[3] Microsoft DirectX. http://www.microsoft.com/windows/directx/default.mspx.

[4] Blepo Computer Vision Library. http://www.ces.clemson.edu/~stb/blepo/.

[5] Stanley T. Birchfield and Daniel K. Gillmor. Fast Bayesian acoustic localization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2002.

[6] Patrick Chiu, Ashutosh Kapuskar, Lynn Wilcox, and Sarah Reitmeier. Meeting capture in a media enriched conference room. In *Proceedings of CoBuild*, October 1999.

[7] Marco Cristani, Manuele Bicego, and Vittorio Murino. Audio-video integration for background modelling. In *Proceedings of the European Conference on Computer Vision*, May 2004.

[8] Ross Cutler, Yong Rui, Anoop Gupta, JJ Cadiz, Ivan Tashev, Li wei He, Alex Colburn, Zhengyou Zhang, Zicheng Liu, and Steve Silverberg. Distributed Meetings: A meeting capture and broadcasting system. In *Proceedings of ACM international conference on Multimedia*, April 2000.

[9] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proceedings of IEEE International Conference on Computer Vision Frame-rate workshop*, Sept 1999.

[10] Ralph Gross, Michael Bett, Hua Yu, Xiaojin Zhu, Yue Pan, Jie Yang, and Alex Waiebel. Towards a multimodal meeting record. In *IEEE International Conference on Multimedia and Expo (III)*, August 2000.

[11] Einat Kidron, Yoav Y. Schechner, and Michael Elad. Pixels that sound. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2005.

[12] Charles H. Knapp and G. Clifford Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, August 1976.

[13] Dar-Shyang Lee, Berna Erol, Jamey Graham, Jonathan J. Hull, and Norihoko Murata. Portable meeting recorder. In *Proceedings of the ACM Conference on Multimedia*, December 2002.

[14] Maurizio Omologo and Piergiorgio Svaizer. Use of the crosspower-spectrum phase in acoustic event location. *IEEE Transactions on Speech and Audio Processing*, 5(3):288–292, 1997.

[15] Heather Richter, Gregory D. Abowd, Werner Geyer, Ludwin Fuchs, Shahrokh Dajjavad, and Steven Poltrock. Integrating meeting capture within a collaborative team environment. In *Proceedings of the International Conference on Ubiquitous Computing*, September 2001.

[16] Stanley J. Rosenschein. Quindi Meeting Companion: A personal meeting-capture tool. In *Proceedings of ACM conference CARPE*, October 2004.

[17] Yong Rui, Anoop Gupta, and JJ Cadiz. Viewing meetings captured by an omnidirectional camera. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, March 2001.

[18] Paul E. Rybski, Fernando de la Torre, Raju Patil, Carlos Vallespi, Manuela Veloso, and Brett Browning. CAMEO: The Camera Assisted Meeting Event Observer. Technical Report CMU-RI-TR-04-07, Robotics Institute, Carnegie Mellon University, January 2004.

[19] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[20] Kevin Wilson, Neal Checka, David Demirdjian, and Trevor Darrell. Audio-video array source seperation for perceptual user interfaces. In *Proceedings of the ACM workshop on Perceptive user interfaces*, November 2001.

[21] Jie Yang, Xiaojin Zhu, Ralph Gross, John Kominek, Yeu Pan, and Alex Waibel. Multimodal people ID for a multimedia meeting browser. In *Proceedings of the ACM Conference on Multimedia*, October 1999.