A STUDY ON PARTICLE FILTER BASED AUDIO-VISUAL FACE TRACKING ON THE AV16.3 DATASET

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

YUNUS EMRE YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

APRIL 2016

Approval of the thesis:

A STUDY ON PARTICLE FILTER BASED AUDIO-VISUAL FACE TRACKING ON THE AV16.3 DATASET

submitted by **YUNUS EMRE YILMAZ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Afşar Saranlı
Supervisor, **Electrical and Electronics Eng. Dept., METU** _____

Examining Committee Members:

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Can Ulaş Doğruer
Mechanical Engineering Department, Hacettepe University _____

Date: **29/04/2016**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: YUNUS EMRE YILMAZ

Signature :

ABSTRACT

A STUDY ON PARTICLE FILTER BASED AUDIO-VISUAL FACE TRACKING ON THE AV16.3 DATASET

Yılmaz, Yunus Emre
M.S., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Afşar Saranlı

April 2016, 107 pages

People tracking has received considerable attention as a research field recently. Since, there are a wide range of application areas that requires to track single or multi target people in different environments with various scenarios using a variety of sensors. In this kind of tracking scenarios, usage of audio and visual information together is commonly preferred method, because these cues are mostly exist in the tracking environment and they contain complementary information about the targets. Our work focuses on particle filter based Bayesian tracking method that fuses location estimates obtained from audio and video data separately for indoor and crowded environments. Surveillance, video-conferencing and security are main examples of application areas for this kind of tracking scenario. In our work, particle filter based trackers are implemented with number of different configurations in order to investigate possible gains from including audio data to the tracking problem instead using only visual data. In these implementations, comprehensive experiments are conducted using the AV16.3 dataset. Usage of this dataset makes possible to compare our results with

other works from the literature. Also, this dataset covers a variety of tracking situations (e.g. occlusions and rapid movements of persons) which can be encountered in realistic scenarios, making the results more useful. Our results indicates that no significant gains are possible when multiple cameras are used except when there are serious optical occlusions.

Keywords: Face Tracking, Audio-Visual Fusion, Particle Filter

ÖZ

PARÇACIK FİLTRESİ TABANLI GÖRSEL-İŞİTSEL YÜZ TAKİBİ SİSTEMİNİN AV16.3 VERİ SETİ KULLANILARAK İNCELENMESİ

Yılmaz, Yunus Emre
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Afşar Saranlı

Nisan 2016 , 107 sayfa

Bir araştırma alanı olarak kişi takibi, son zamanlarda kayda değer miktarda dikkat çekmektedir. Çünkü; tek ya da fazla sayıda kişinin hedef olarak seçildiği, değişik senaryolar içerisinde farklı özelliklere sahip sensörlerin de kullanıldığı bir çok uygulama alanında, kişi takibine ihtiyaç duyulmaktadır. Bu tarz takip senaryolarında, ses ve görüntü verilerinin birlikte kullanılması oldukça tercih edilen bir yöntemdir, zira bu veriler takip alanında hâlihazırda bulunmakta ve birbirleri ile tamamlayıcı bilgiler içermektedirler. Çalışmamızda, ses ve görüntü verilerini ayrı ayrı kullanarak kapalı ve kalabalık mekanlarda konum tahmininde bulunan Bayes teoremine dayalı parçacık filtresine odaklandık. Gözetleme, video-konferans ve güvenlik, bu tarz takip sistemlerinin en temel uygulama alanlarıdır. Çalışmamızda; parçacık filtresine dayalı takip sistemi, değişik düzenleme biçimleri ile sadece görüntü verisi yerine ses verisinin de sisteme eklendiği durumlardaki kazancı inceleyebilmek amacı ile gerçeklenmiştir. Bu gerçeklemeler sırasında, kapsamlı deneyler AV16.3 veri seti kullanılarak yapıl-

mıştır. Bu veri setinin kullanımı ise yapılan işin sonuçlarını literatürdeki diğer işlerle karşılaştırma imkanı yaratmaktadır. Ayrıca, bu veri seti değişik gerçekçi senaryoları da(hedefin veya hedeflerin görsel olarak başka cisim veya cisimler tarafından engellenmesi ve hedef kişilerin ani hareketi gibi durumları) kapsayarak sonuçların daha faydalı olmasını sağlamıştır. Çalışmamızın sonuçları göstermektedir ki çoklu kameranın kullanıldığı durumlarda, eğer ciddi bir görsel engelleme yoksa, ses verisinin eklenmesinin ciddi bir katkısı olmamaktadır.

Anahtar Kelimeler: Yüz Takibi, Görsel-İşitsel Data Birleştirme, Parçacık Filtresi

To my family

ACKNOWLEDGMENTS

I would like to thank my supervisor Professor Afşar Saranlı for his constant support, guidance and friendship. It was a great honor to work with him for the last two years and our cooperation influenced my academical view highly.

I would like to thank Dr. Handan Ağırman for her guidance at the beginning of this thesis and her reviews at end of this thesis. Her inspiring ideas help me to form the content of the thesis.

I would like to acknowledge the support of ASELSAN Inc. during the thesis.

My family and my friends also provided invaluable support for this work. I would like to thank them all.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ASELSAN	(Turkish) Askeri Elektronik Sanayi, Military Electronic Industries
AV	Audio-Visual
AV-PF	Audio-Visual Particle Filter from [18]
AV-PF-RAND	Particle Filter Based Audio-Visual Fusion Technique in 3-D by Using Two Cameras and Two Microphone Arrays with Occlusion Handling
AV-PF-3D	Particle Filter Based Audio-Visual Tracking Technique Using Two Cameras and Two Microphone Arrays
AV-PF-2D	Particle Filter Based Audio-Visual Tracking Technique
AV-PF-2D-SNS	Particle Filter based Audio-Visual Tracking Technique with speech/non-speech classification
CCD	Charged-Coupled Devices
CMOS	Complementary Metal–Oxide–Semiconductor
CRF	Conditional Random Field
DBN	Dynamic Bayesian Network
DOA	Direction of Arrival
EKF	Extended Kalman Filter
FPS	Frame per Second
HMM	Hidden Markov Model
LPC	Linear Predictive Coding
MAE	Mean Absolute Error
MFCC	Mel-frequency Cepstral Coefficient
MİKES	(Turkish) Mikrodalga Elektronik Sistemler, Microwave Electronic Systems
PF	Particle Filter
R&D	Research and Development
SRP-PHAT	Steered Response Power Phase Transform
SSE	Sum of Squares due to Error
SSM	Sam-Spare-Mean
SVMs	Support Vector Machines

3-D	Three dimensional
TDOA	Time Difference of Arrival
2-D	Two dimensional
UCA	Uniform Circular Array
UKF	Unscented Kalman Filter
V-PF	Visual Particle Filter from [18]
V-PF-2CAM	Particle Filter Based Visual Tracking Technique in 3-D by Using Two Cameras

LIST OF VARIABLES

$\mathbf{x}$	bold lower case denotes vectors
N	the number of particles
$n = 1, ..., N$	the particle index
K	the total number of image frames
$k = 1, ..., K$	the image frame index
$\tilde{\mathbf{x}}_k$	estimated target position at the frame k
$\tilde{\mathbf{x}}_k^n$	the position of the n-th particle at the frame k after incorporating DOA
$\mathbf{F}$	bold capital denotes matrices
$\|.\|_1$	l_1 norm
$\odot$	the element-wise product
$\oplus$	the element-wise addition

CHAPTER 1

INTRODUCTION

1.1 Motivation

In an indoor and crowded environment, one or more targets can be needed to be tracked due to various reasons. Tracking in this context means capturing the face image of the target person, while listening to her/him. It is obvious that the view and the sound of the speaker are not always available at the same time. In other words, the target can be occluded or can be silent for a while. Hence, the tracker could handle these kind of challenges in the lack of one data type or both. For example, a target person is required to be tracked during a seminar or meeting in a conference room due to some security related reasons. Another example is the tracking of basketball player during the match. By tracking the field location of the player with the ball would allow the camera to automatically follow the player, hence TV viewers would have a more vivid watching experience. Apparently, the examples about these kind of tracking systems can be augmented. The present thesis focuses on particle filter based tracking methods as a solution to these kind of systems.

The motivation of the thesis lies in a system built in MİKES, later acquired by ASELSAN, and named as "Sound Detection and Analysis System for Far-Field and Near-Field Sources" [1]. The side view of that acoustic beamforming system can be seen in Figure 1.1. Final product of this project was an array system consisting of 255 microphones with a sophisticated digital processor card. Briefly, that acoustic system can create 5 different beams in order to listen desired directions given in azimuth and elevation angles with respect to red dotted center shown in Figure 1.1, while suppress-

ing sound and noises from other directions. The initial aim of the system was only listening the desired directions using beamforming. However, the research about the localization of speaker/speakers was in progress.

Figure 1.1: Acoustic Beamforming System from Side View. Red dot on the Figure shows the center of the microphone array.

However, this system is not automated. In other words, it always needed an human operator to enter the angles of the target speaker. As the source of the speaker, mouth locations of the speakers are required to be listened. However, it is not easy to them with respect to the center of the system using angkes of the spherical coordinate system. In order to automate this system, we proposed to add a visual sub-system to it. That has constituted the initial focus of the thesis as it is illustrated in Figure 1.2.

That newly envisioned system formed a good motivation for us to implement a tracker system fusing the audio and the visual data. Unfortunately, that project has been stopped with the acquisition of MİKES and hence, the equipment became unreachable for us. The thesis focus was re-adjusted to include the existing AV16.3 dataset[23] as the basis of our investigations.

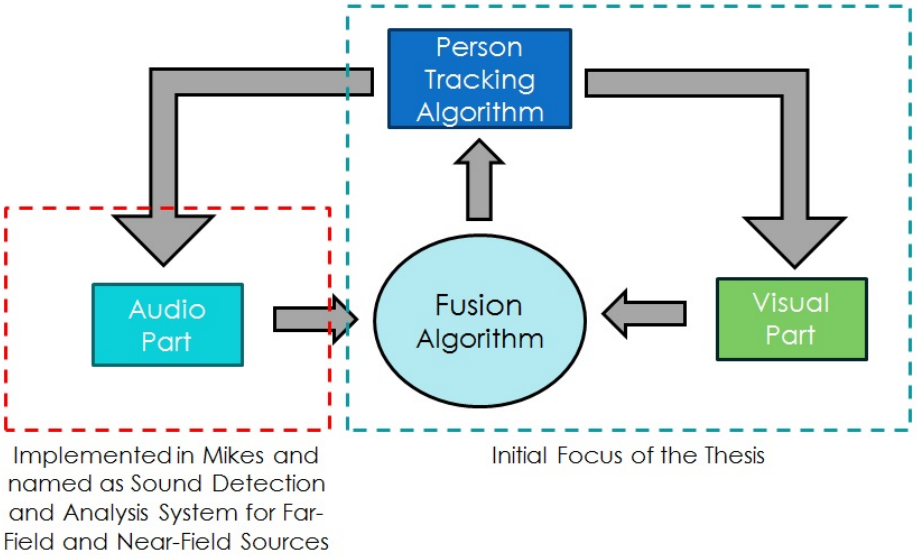


Figure 1.2: Initial Focus of the Thesis.

1.2 Multi-Person Tracking

1.2.1 Audio-Visual Approach as Multimodal Fusion

Multimodal fusion is one of the contemporary research areas, since it's beneficial in various multimedia analysis tasks. Any stage in the integration process can be referred as $multimodal\ fusion$, if the different sources of data are combined [25]. By processing the multimodal data in the fusion stage, beneficial insights about the data, a situation or a higher level of activity can be obtained [2]. Semantic concept detection, audio-visual speaker detection, event detection or human tracking, as in our case, are main examples of the multimedia analysis.

Audio-visual fusion is a specific case of multimodal fusion analysis. Audio and visual data are input sources as its name indicates. Usage of these two input sources are beneficial since they are correlated and convey complimentary information[2]. For example, in our case, the mouth of the speaker can be localized using the audio data or the video data separately and usage of these two increases the overall accuracy of the tracker.

The system in this thesis is designed to track faces of the targets in a stochastic dynamical system. By considering the fact that the camera measurement model is nonlinear and the the motion of the face is also nonlinear, particle filter approach is chosen to be implemented as a solution to the tracking problem.

1.2.2 Particle Filter Approach

The estimation based techniques are basically variants of the Kalman filter methods and particle filter methods.

Kalman filtering is an optimal state-space estimation method for measurements of linear systems. Additionally, Kalman filtering assumes Gaussian noise on the measurements. Also, for nonlinear systems, Extended Kalman filter or Unscented Kalman filter can be used.

Particle filter is an another approach for modeling stochastic systems. Contrary to usual Kalman filter, particle filters are more suitable for nonlinear systems and it assumes non-Gaussian noise on the measurements.

Particle filtering has become an established technique for modeling stochastic dynamical systems. Particle filter methods are based on Monte Carlo techniques. Although these techniques have existed since the 1950s[16], they are disregarded due to lack of computational power at the time and problems with degeneracy. However, since the bootstrap filter [15] and more general resampling schemes are proposed, the research in this area has rapidly increased[33].

As it is indicated before, in our case, the audio and the video data are fused in the particle filter framework for face tracking of the target people.

1.2.3 Face Tracking Techniques Based on Particle Filtering Method Using Audio-Visual Fusion

In this thesis, specifically, face tracking of multiple moving person in a close and crowded environment such as seminar room or smart room are examined. The envi-

ronments of the earlier techniques were static. Also, these one person techniques were designed for controlled environments. Increasing processing power with algorithmic and theoretical progresses have caused that more advanced methods have developed for multiple speakers in a dynamic and natural environments. Furthermore, initially, single modality sensor types were used. Nevertheless, types of sensors are evolved to multi-modality [18].

Although, in the literature, there are video only and audio only tracking algorithms, their performance are limited. Video only algorithms suffer from the occlusion case, while performance of audio only algorithms are limited due to intermittent nature of audio data, background noise and reverberation of the indoor environment. Hence, audio and visual information can be fused for a tracking system in a way that overall performance is better than any single-modality based tracking system.

After conducting a literature survey and considering the tracking system depicted in Section 1.1 and Figure 1.2, it is decided that particle filter based audio-visual tracking systems are good candidates to solve the described tracking problem.

1.3 Thesis Contribution

Contribution of the thesis can be concluded in three main parts.

- From starting a simple and standard 2-D particle filtering approach by using one camera and two microphone arrays for audio-visual fusion, a 3-D tracking system with the two camera and two microphone arrays are implemented. Also, all of the tracking results of the intermediate steps are presented.

- Results of the changes between the methods are analyzed and these analyzes are presented to show improvements or drawbacks of the tracker systems.

- It is shown that with low cost microphones and cameras, 3-D face tracking system for multiple target can be implemented.

1.4 Thesis Organization

As it is stated, this works focuses on particle filter based audio-visual face tracker. Hence, the chapters are organized to present detailed information about literature survey and implementation details.

Chapter 2 contains the literature survey about the topics of the model of audio sensors, the model of visual sensors, the audio-visual fusion and particle filtering given through the thesis.

Chapter 3 consists of detailed information about the AV16.3 dataset containing synchronized video and audio data used in this thesis.

Chapter 4 contains the detailed explanation of eight methods implemented in the thesis.

Chapter 5 presents the results of implementation. Additionally, analyzes and comparisons of the results are demonstrated.

Chapter 6 concludes what is implemented and analyzed in this thesis, while that chapters presents also the future works.

CHAPTER 2

AUDIO-VISUAL FUSION BASED ON PARTICLE FILTERS

2.1 Introduction

The focus of this thesis is particle filter based audio-visual fusion. In this chapter, background information about this topic is presented. For that purpose, firstly, used audio and visual sensors are described. These sensors are implemented to estimate the location of the target separately. Hence, techniques of localization is explained for both. After that part, audio-visual fusion is depicted. Lastly, particle filtering framework where the audio-visual fusion takes place is explained.

2.2 Modeling of Audio Sensor

In this thesis, as it is indicated, microphone arrays are used as audio sensors. Basically, a microphone can be described as a sensor that converts sound signal to electrical signal. For localization, a set of microphone is needed, since the correlation between the signals gives information about the location of the audio source. In this thesis, localization approach of Lathoud[22] is used. Also, AV16.3 Dataset used for simulation. Related information about this dataset is presented in Chapter 3 in a detailed way.

The approach in [22] is sam-spare-mean(SSM) method and that method includes two step. The first step is the sector-based detection and localization. Microphone array with 8 microphones $\{l_1...l_8\}$ is placed in a circular manner. These 8 microphones divides space around them into 18 sectors $\{S_1...S_{18}\}$. Microphone configuration and

sectors are illustrated in Figure 2.1. Firstly, an "activeness" measure is evaluated at each frame for each sector. After that, a binary decision is made by comparing that activeness measurement with predefined threshold.

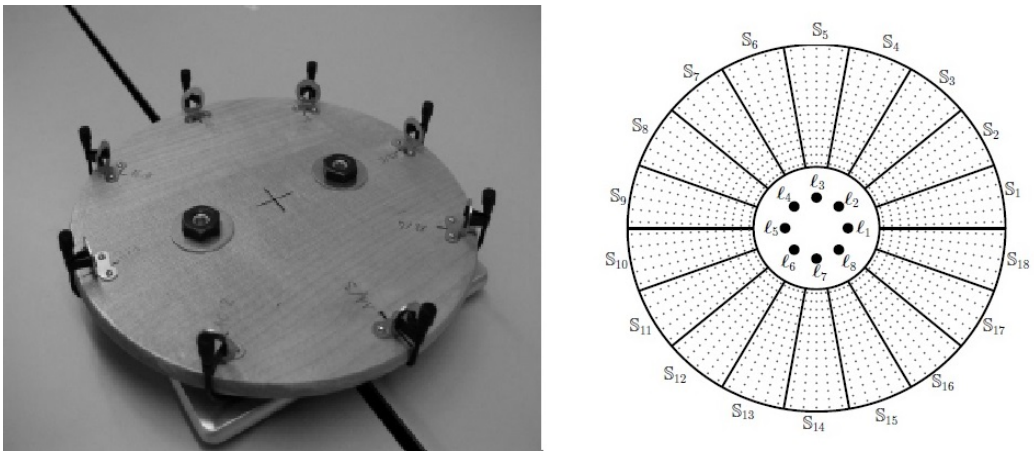


Figure 2.1: The microphone array with 8 microphones $\{l_1...l_8\}$ divided into 18 sectors $\{S_1...S_{18}\}$ [21].

In the second step, for the sectors having at least one active source, a parametric point-based search is operated for localization. Parameters in this search are optimized with respect to the cost function which is SRP-PHAT in our case. Derivation of the DOA angle can be found in [21] in a more detailed way. Furthermore, the illustration of these two steps is given in Figure 2.2.

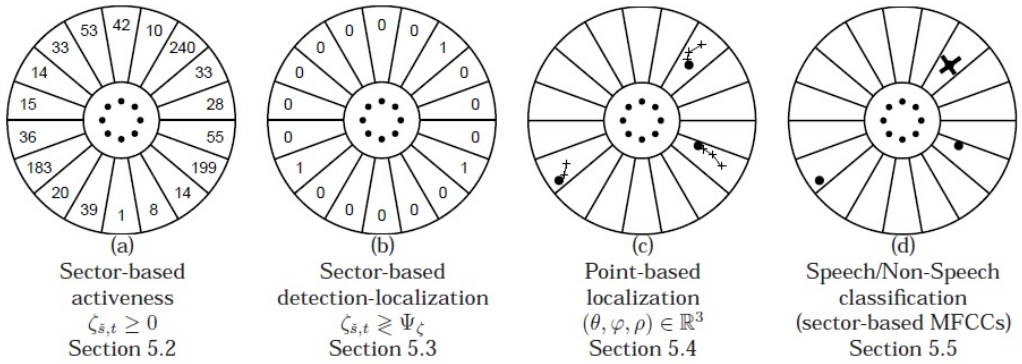


Figure 2.2: Proposed multisource detection-localization. The eight dots in the center represent the microphone array. The three dots in the sectors represent point location estimates [21].

All the steps implemented in this audio localization technique is summarized in Figure 2.3.

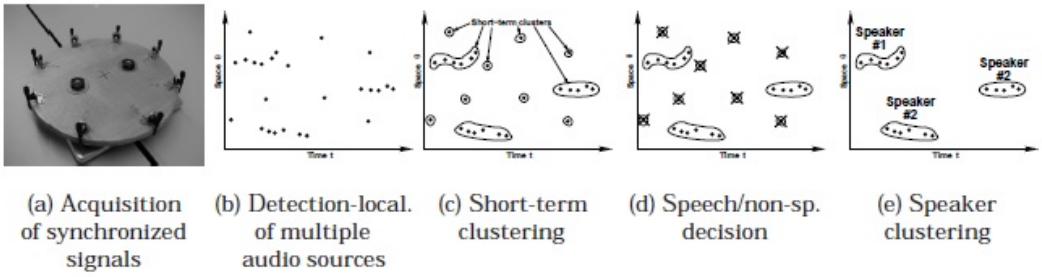


Figure 2.3: All the steps implemented in this audio localization of [21].

2.3 Modeling of Visual Sensor

2.3.1 Image Formation

In order to explain the camera model in this thesis, firstly, the simplified pinhole model shown Figure 2.4 should be expressed. In this simplified model, a box with a small hole in center of one of its sides and a semitransparent plate on the opposite side is used. If some light rays from an object faces that box from the side with the small hole, then the inverted image of the object will appear on the semitransparent plane[11]. For instance, the case with a candle is illustrated in Figure 2.4.

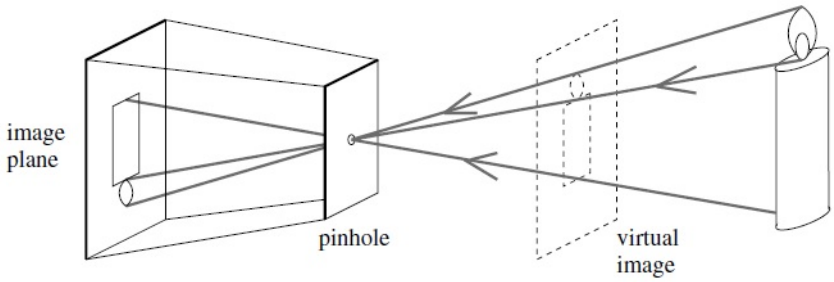


Figure 2.4: The pinhole imaging model [11].

Despite the physical impossibility, if the pinhole is assumed to be a point, then each

light ray from the scene point must travel to the semitransparent plane, named as image plane, by passing through the pinhole point. However, in reality, the size of the pinhole is small but finite. Hence, each point on the inversed image shown in Figure 2.4 is the intersection of light rays coming from related solid angle. Consequently, that simple and idealized model doesn't fit the real world scenarios. Furthermore, instead of small pinholes, real cameras are equipped with lenses. Lenses are more useful for gathering the light rays, but they complicates the model due to physical restrictions of the lenses. Moreover, image planes are equipped with CMOS or CCD sensors to capture light rays for image formation. Despite its idealized assumptions and simplicity, the pinhole approach is mathematically convenient. Additionally, it also presents adequate approximation of the imaging process [11].

In the following section, by considering the real case scenario with the lens, the camera calibration procedure is explained.

2.3.2 Camera Calibration

A real camera model in its simplest form is consisted of a pinhole and an image plane. Location of the pinhole is between the observed 3-D world scene and the image plane.

As it is stated in the previous section, light rays emitted or reflected from a surface of any object must travel through the pinhole before arriving the image plane. Consequently, it can be stated that each 2-D area or point in the image plane corresponds to an area in the 3-D world. That projection mechanism is the explanation of the image formation. In Figure 2.5, a mathematical model of that model is shown. By using that model, the concept of the projection is simplified to the concept of the magnification. There are two important systems that should be understood so that the relation between the points in the real world and their image plane correspondences can be described[10]:

1. The external coordinate system. In Figure 2.5, 'world' is denoted with 'W'. The placement of that coordinate systems and parameters used to describe it are independent from the camera.
2. The camera coordinate system. In Figure 2.5, 'camera' is denoted with 'C'.

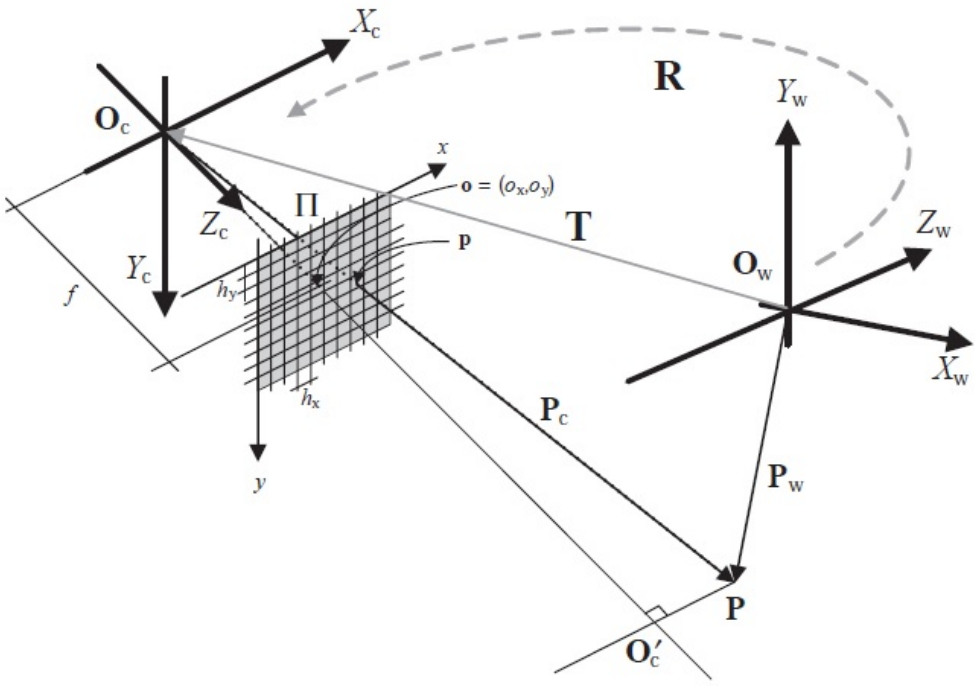


Figure 2.5: Pin-hole model of the perspective camera. Two separate coordinate system W and C illustrates the world and camera coordinate systems respectively[10].

In order to explain the relation between W and C, two matrix should be defined: a translation matrix **T**, and rotation by matrix **R**.

$\mathbf{O_c}$ is named as the $central$ or $focal\ point$ in the camera coordinate system constituted of X_c, Y_c and Z_c. Image plane $\mathbf{\Pi}$ is a critical part of the camera model and as it can been observed in Figure 2.5, it is composed of rectangular pixel elements. For indexing the each of these pixels, pair of natural numbers are used. The location of *the principal point*(o_x, o_y) is the projection of $\mathbf{O_c}$ on the image plane $\mathbf{\Pi}$ in the direction of Z_c. The line between $\mathbf{O'_c}$ and $\mathbf{O_c}$ is named as $principal\ axis$. Also, the distance between the image plane and the principal point is named as $focal\ length$. Additionally, the h_x and h_y given in the model are physical dimensions of a single pixel.

Point **P** in the space can be denoted as:

- $\mathbf{P_w} = [\ X_w\ Y_w\ Z_w\ 1\ ]^T$ in the 3-D external world (W) homogeneous coordinate

system.

- $\mathbf{P_c} = [\ X_c\ Y_c\ Z_c\ 1\ ]^T$ in the 3-D internal camera (C) homogeneous coordinate system.
- $\mathbf{P_p} = [\ P_p\ Y_p\ Z_p\ 1\ ]^T$ in the 2-D pixel camera ($\mathbf{\Pi}$) homogeneous coordinate system.

In order to relate $\mathbf{P_w}$, $\mathbf{P_c}$, and $\mathbf{P_p}$, two types of parameters are needed:

1. Extrinsic Parameters
2. Intrinsic Parameters

For a rotation matrix $\mathbf{R}$, translation matrix $\mathbf{T}$ and internal camera calibration matrix $\mathbf{K}$, relations between coordinate system is shown below:

- $\mathbf{P_c} = [\ \mathbf{R} \mid \mathbf{T}\ ] * \mathbf{P_w}$
- $\mathbf{P_p} = \mathbf{K} * \mathbf{P_c}$
- $\mathbf{P_p} = \mathbf{K} * [\ \mathbf{R} \mid \mathbf{T}\ ] * \mathbf{P_w}$

These matrices and their contents will be explain in the following two sections.

1) ***Extrinsic Parameters***

$[\ \mathbf{R} \mid \mathbf{T}\ ]$ is a 4x4 matrix in homogeneous coordinate system and it is named as external calibration matrix. That matrix is combination of 4x3 rotation matrix $\mathbf{R}$ and 4x1 translation matrix $\mathbf{T}$.

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \\ 0 & 0 & 0 \end{bmatrix} \mathbf{T} = \begin{bmatrix} T_{11} \\ T_{21} \\ T_{31} \\ 1 \end{bmatrix}$$

2) ***Intrinsic Parameters***

Intrinsic parameters are used to model transformation from 3-D camera coordinates including distortions due to physical constraints of the camera. In the homogeneous coordinates, internal calibration matrix $\mathbf{K}$ is [37]:

$$\mathbf{K} = \begin{bmatrix} \alpha & c & o_x & 0 \\ 0 & \beta & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

In $\mathbf{K}$ matrix, α and β represent the scale factors of the image in x-axis and y-axis in the image plane $\mathbf{\Pi}$. Coordinates of the principal point in x-axis and y-axis are (o_x, o_y) respectively. Also, the skewness of the two image axes is denoted by c[37].

Instead of providing the extrinsic and intrinsic parameters separately, a single projection matrix $\mathbf{P}_{proj}$ as the product of these two matrices can be provided.

$$\mathbf{P}_{proj} = \mathbf{K} * [\mathbf{R}|\mathbf{T}]$$

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} = \begin{bmatrix} \alpha & c & o_x & 0 \\ 0 & \beta & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_{11} \\ R_{21} & R_{22} & R_{23} & T_{21} \\ R_{31} & R_{32} & R_{33} & T_{31} \\ 0 & 0 & 0 & \end{bmatrix}$$

In AV16.3 dataset [23], the projection matrices for each camera are provided.

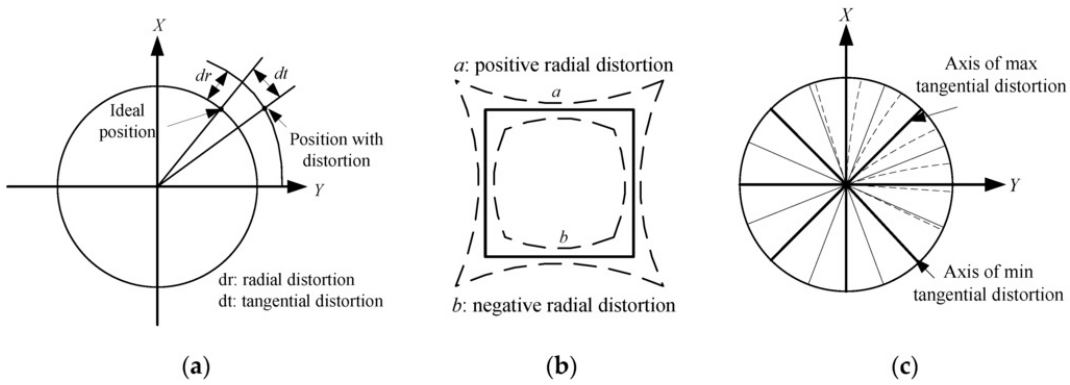
Lens distortions are also another error source for the image formation. Brown-Conrady model suggested by Brown in 1966 [8] is generally known model for lens distortions. The radial and tangential distortions are shown in Figure 2.6.

Lens distortion can be calculated as below[36]. For the calculations, normalized pixel locations(x_n, y_n) with respect to principal points in the image pixel plane $\mathbf{\Pi}$ are used.

Total distortion δ due to the lens is the summation of the radial and tangential distortion.

$\delta = \epsilon_{rad} + \epsilon_{tang}$

Radial distortion ϵ_{rad} can be calculated as:



(a) (b) (c)

Figure 2.6: (a) Radial and tangential distortions; (b) Effect of radial distortion; (c) Effect of tangential distortion. [36].

$$\epsilon_{rad} = (1 + K_1 * r^2 + K_2 * r^4 + K_3 * r^6) \begin{bmatrix} x_n \\ y_n \end{bmatrix} \tag{2.1}$$

K_1, K_2, K_3 : Radial distortion coefficients

$r^2 = x_n^2 + y_n^2$

Also, the tangential distortion ϵ_{tang} can be calculated as:

$$\epsilon_{tang} = \begin{bmatrix} 2 * K_4 * x_n * y_n + K_5 * (r^2 + 2 * x_n^2) \\ K_4 * (r^2 + 2 * y_n^2) + 2 * K_5 * x_n * y_n \end{bmatrix} \tag{2.2}$$

K_4, K_5 : Tangential distortion coefficients

After the explanation of the audio and visual sensor models, next section continues with the general description and the explanation of the stages and types of the audio-visual fusion.

2.4 Audio-Visual Fusion

2.4.1 Description

Nature of the human interaction and activity is multimodal. Vision and hearing are primary senses to discern the complex outside world [30]. Different information

about the scene can be obtained using audio and video modalities. Video signals contain the information about appearance such as color, texture, shape and the distribution of the objects in the scene. In addition to video signals, audio signals include information about the sound such as speech, music and noise [9]. An example of the video and related audio signal is shown Figure 2.7. People link the visual and audio cues intiutively. For example, the relationship between a falling object and the sound of the smash can be easily understood, moving lips to the presence of speech can intuitively linked, or it is known that what kind of music will be heard when a guitarist's arm moving. Consequently, in order to better understand a scene, audio and video signals should be jointly processed rather than considering each modality separately [9].

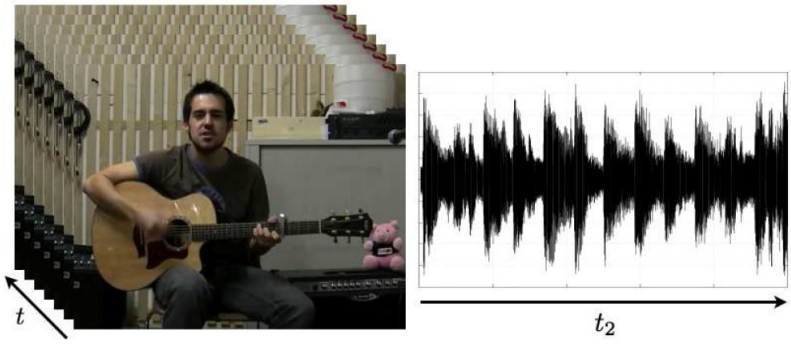


Figure 2.7: Example of a 3D video signal [left] and the corresponding 1D audio signal [right]. The temporal axis of each modality presents a different resolution [9].

There are a lot of areas that audiovisual fusion can be used [17]: active speaker localization and tracking, bio-metrics verification, concept detection, emotion recognition, event detection, human or object tracking, human–computer interaction, meeting segmentation, monologue detection, music content analysis, speaker recognition, speech recognition, source separation, story segmentation in news video, video retrieval, video shot detection, and voice activity detection. From those listed areas, as it is stated previously, the focus of the thesis is the human tracking.

A basic AV process consists of two main steps [17]. First step is feature extraction from each modality with respect to the application and the second step is the integration of the information conveyed by the modalities. Generic representations of these

two main steps are shown in Figure 2.8 and Figure 2.9.

2.4.2 Feature Extraction

The feature extraction part depends on the application and the modalities. Before the fusion stage, representing audio and video modalities in a convenient and an effective feature space is an important step. Fortunately, audio sources have some well known representative features. Yet, visual features are not well-defined. For the audio part, spectrum-based features, like MFCCs(Mel-frequency cepstral coefficients) and LPC(linear predictive coding), prosodic features, phoneme posterior features are well known examples of these features. In general, which informative parts of the body will be used in the visual tracking are based on the application. For example, these parts can be mouth or eye regions. In our case, head figures7 of the targets are informative parts. [17]. The feature extraction step is shown in Figure 2.8.

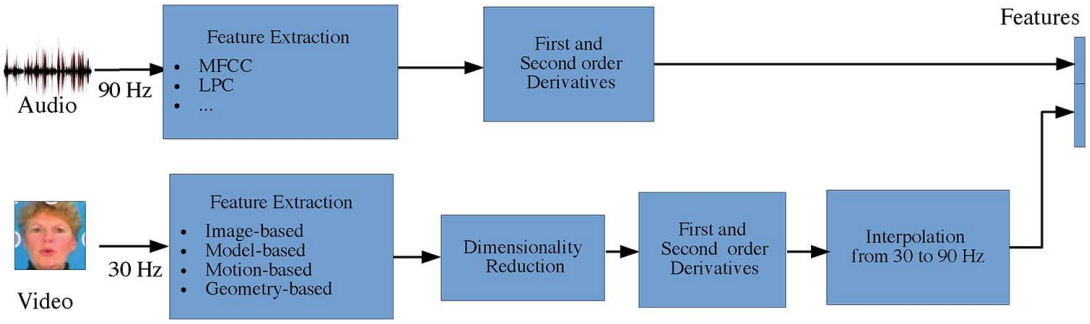


Figure 2.8: Generic scheme representation of a feature extraction system for an audio-visual fusion [17].

2.4.3 Integration of Modalities

In the traditional manner, fusion approaches are classified under three main sections, namely early, late and intermediate integration. The fusion stage can be operated in any level of the data integration. If modalities are fused before by integrating and combining modalities of the features, it is named as *early integration*. Contrary to the early integration, if the separate model of each modality is obtained before the integration of the final decisions and all these modalities are fused to generate

final decision, then it is named as *late integration*. Addition to these two methods, if the fusion is performed between early and the late integrations, it is named as *intermediate integration*. In some sources, intermediate integration is also called as early integration. Lastly, by combining the fusion at early and late levels, *hybrid approach* can be obtained. [17] Both early and late integration block schemes are shown in Figure 2.8.

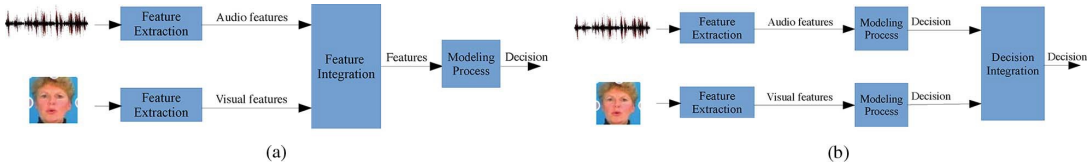


Figure 2.9: (a) Early integration. (b) Late integration [17].

In addition to the traditional manner, in the survey of Shivappa *et al.* [30], fusion approaches are classified with respect to their intents. It is stated that "intent" is important, since the researchers address it in designing a system. In the Figure 2.10, data fusion at different levels of signal abstraction and the fusion level of our work is illustrated. The original diagram is modified so that our work can be shown on it with blue rectangles.

By comparing the traditional manner and this new approach, in diagram shown in Figure 2.10, signal extraction level and semantic level are defined as new fusion levels. In the new diagram, the early integration corresponds to feature level fusion, while the late integration corresponds to the decision level fusion. In addition to these, the intermediate fusion in the traditional manner can be correlated with classifier level fusion. With respect to traditional manner the implemented techniques in this work can listed under early integration techniques. Hence, considering the new approach they can be listed under feature level fusion.

2.4.4 Fusion Techniques

In the literature, by considering the modelling and the integration properties of the approaches in the AV processing, the AV fusion techniques are classified under five

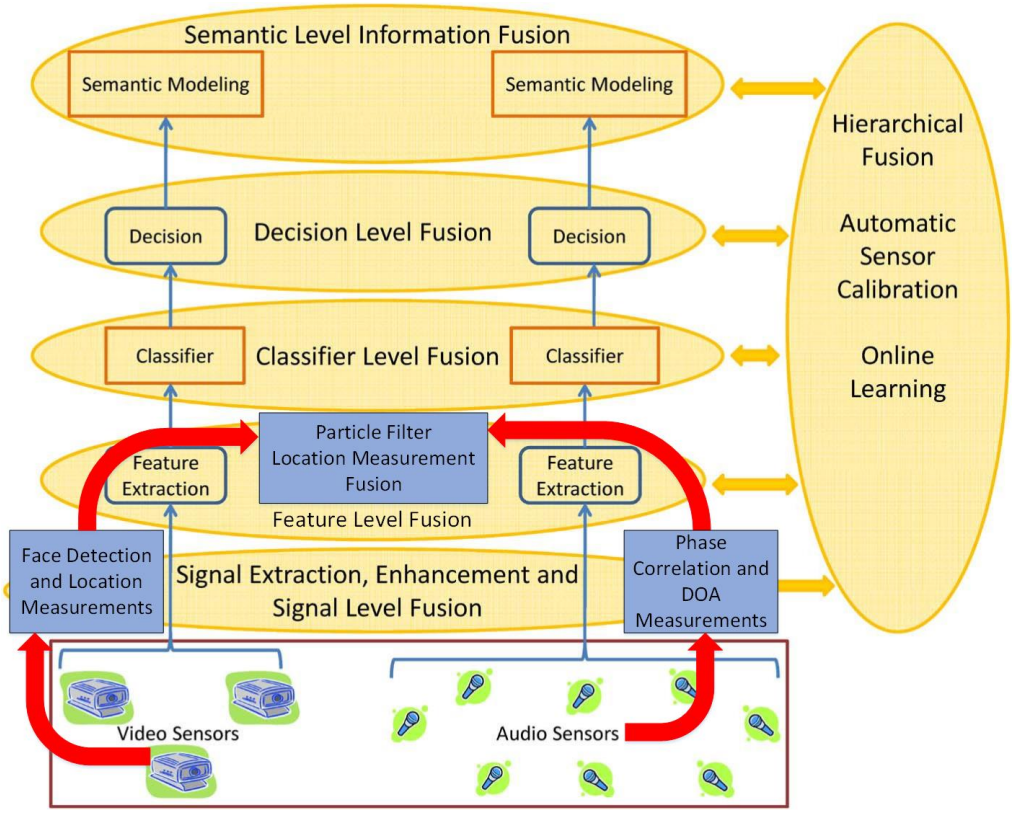


Figure 2.10: Data fusion at different level of signal abstraction and the data fusion level of the implemented methods in the thesis [30].

main methods [17]:

1. Support Vector Machines
2. Dynamic Bayesian Networks
3. Hidden Markov Models
4. Estimation-Based Methods
5. Task-Dependent Techniques

The first three methods are mostly related with the modeling part. However, the modeling of AV processes is not in the scope of this thesis. The scope is on the state estimation and tracking part. Hence, only the descriptions and their relations to the state estimation and tracking are briefly explained in the following parts for DBNs and HMMs.

- ***Dynamic Bayesian Networks:***

  In order to illustrate a set of random variables with their conditional dependencies, probabilistic graphical models are used. These graphical models are called Bayesian networks. In these illustrations, Bayesian networks are constructed with acyclic directed graphs. Each variable in this representation is represented with a vertex, and an edge between the corresponding vertices depicts the conditional dependency between two variables. DBN is a sub-technique in Bayesian networks and this technique is used to model sequences of observations [17]. The example illustration of the DBN is shown in Figure 2.11. In AV applications, particularly where temporal sequencing is considered, DBNs are mostly preferred. Speech processing and video analysis are two examples of the temporal sequencing.

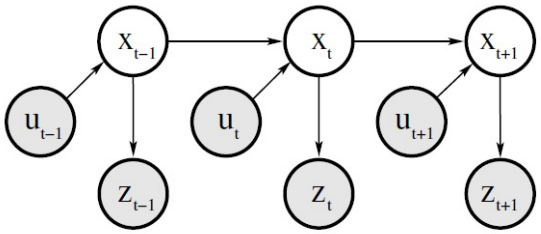


Figure 2.11: An example of Dynamic Bayesian Network [32].

  DBNs can be preferred in AV fusion task which needs the dependencies between their random number to be determined. In addition to that, DBNs provide an efficient way to manage the time-series data. Hence, they are advantageous for analysis tasks in multimedia applications.

- ***Hidden Markov Models:***

  For the representation of probability distributions over sequences of observations, HMMs, as the simple form of the DBN, are used. HMMs are also popular method for multimedia analysis like DBNs. A single HMM can be used to represent AV features jointly, without discriminating between them in some works and these works can listed under the early integration techniques. For instance, in the work of Wang $et\ al.$ [35], AV features are extracted from each frame

to execute video shot detection by using HMM. Furthermore, several different versions of HMM are presented as the intermediate integration approaches. In these techniques, separate modalities are tried to be represented while their interactions are examined at the same time.

In addition to HMMs and DBNs, other graphical models, e.g. CRFs conditional random fields, and their variations have been utilized for multimodel fusion.

- ***Estimation-Based Methods:***

  Estimation-based techniques are extensively used in the fusion of the multiple sources and these techniques comprise Kalman filters, particle filter and their variants. In the Kalman filtering, the state-space model is constituted by observing noisy data sequence over time. Due to its ability to retain its previous states, Kalman filter needs no extra memory for the storage of the past. For linear systems containing additive Gaussian noise, Kalman filtering is optimal. Furthermore, for estimating the states of the nonlinear systems, Extended Kalman Filter(EKF) or Unscented Kalman Filter(UKF) can be used.

  Particle filters are utilized to estimate the states of the stochastic dynamical systems by observing the series of data over time. As it is stated, basic Kalman filter is used for linear systems with additive Gaussian noise, while EKF and UKF are used with nonlinear systems. However, particle filter is more appropriate to use with nonlinear systems with non-Gaussian noise.

  These two techniques are popular data fusion, object/people localization and tracking. With respect to the application area, these two techniques can be applied at both feature or decision levels of the fusion. For instance, Loh $et\ al.$ [24] used three microphones to record the audio data and one camera to collect the visual data. These two data are fused to estimate the position of the speaker. After the fusion stage, velocity and acceleration of the speaker are estimated by a Kalman filter. Gehring $et\ al.$ [14] implemented a technique that uses recognized faces from different cameras as video feature. Also, TDOA(time delay of the arrival) between different microphones are calculated as audio feature. These features are combined in EKF for the localization of the active speaker. In addition to these, Talantzis $et\ al.$ [31] presented a hierarchical Kalman filter structure with multiple microphones and cameras so that implemented filter

tracks people in 3-D space. Firstly, two separated local Kalman filters are implemented to gather audio and video data. After collecting data, the outputs of these two local filters are combined in one global Kalman filter.

Kılıç *et al.* [18] implemented an audio assisted visual tracker with an adaptive particle filter. In that work, firstly, a visual-only particle filter is implemented. Second method contains the audio integration to the first tracker. Lastly, the final form of the tracker is adaptive version of the second one. In these methods, one camera collects the visual data, while an array of eight microphones collects the audio data and multiple targets are tracked on 2-D image plane. At the final tracker, in the propagation step, the Gaussian noise distribution of particles is reshaped. In the measurement step, the observation model is reweighted by using the audio information and DOA(direction of arrival angle). In this thesis, first two methods are implemented as recent methods for comparison. Also, the details of these two trackers are explained in Section 4.2 and 4.3.

- ***Task-Dependent Techniques:***

  In the literature, there are additional techniques for specific applications. However, these methods have no general applicability. Furthermore, these techniques are mostly listed under intermediate integration techniques.

Due to applicability to nonlinear system with non-Gaussian noise, ability to represent arbitrary densities, and capability of tracking the maneuvering multiple targets, particle filter is reasonable choice the tracking system described in Section 1.1. Concepts of particle filtering and implemented algorithms in this thesis are investigated in the following section.

2.5 Particle Filters

2.5.1 Introduction

Particle filter is a nonparametric type Bayes filter. Using that filter, state estimates for the stochastic dynamical systems are obtained based on recursive observations in time. That filtering approach based on sequential importance resampling and

Bayesian theory, these concepts are examined in the following sections. It is a powerful method for non-linear systems with additive non-Gaussian noise.

Before going into the details of the particle filter algorithm, it is proper to examine basic Bayes filtering algorithm.

2.5.2 Basic Algorithm of Bayes Filtering

The $belief$ concept is used to reflect the knowledge about the state of the system, since the true state can not be measured directly in stochastic dynamical systems. In the literature, the belief is also named as *information state* and *state of knowledge*. Conditional probabilities represent the belief distribution which allocates a probability or density value to each possible hypothesis concerning the true state.

Belief distributions are posterior probabilities over state variables determined on the existing data. By denoting the state variable as x_t at time t the belief $bel(x_t)$ is calculated as:

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \tag{2.3}$$

This formula shows the relation of the posterior probability distribution $bel(x_t)$ calculation using the the state x_t at time t which is conditioned on all past measurements $bel(z_{1:t})$ and all past controls $bel(u_{1:t})$.

It is useful to calculate a posterior before incorporating the measurement z_t and just after performing the control u_t. This posterior is called as $prediction$ and denoted as:

$$\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t}) \tag{2.4}$$

$\overline{bel}(x_t)$ predicts the state at time t, before considering the measurement at time t. The $belief$ distribution can be calculated from the $prediction$ and that is called $correction$ or $measurement\ update$.

By using $prediction$ distribution, The pseudo-code of the basic Bayesian algorithm is given in Table 2.1.

As it shown in the pseudo-code, it is a recursive algorithm. The previous belief

Table2.1: The general algorithm for Bayes filtering [32]

1:	**Algorithm Bayes Filter(**$bel(x_{t-1}), u_t, z_t$**)**
2:	$for\ all\ x_t\ do$
3:	$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\ bel(x_{t-1})\ dx_{t-1}$
4:	$bel(x_t) = \eta\ p(z_t \mid x_t)\ \overline{bel}(x_t)$
5:	$end\ for$
6:	$return\ bel(x_t)$

$bel(x_{t-1})$, control u_t and measurement z_t are inputs of the algorithm. Output of the algorithm is the belief at the current time $bel(x_t)$. Line 3 shows the calculation of the prediction using previous belief. Also, Line 4 represents the calculation of the current belief using the prediction and this stage is named as the correction.

Next section describes the particle filters using the explanations about the belief in this part.

2.5.3 Basic Algorithm of Particle Filtering

The posterior distribution or belief is approximated by a finite number of parameters in all nonparametric Bayes Filters. However, with respect to the chosen method, producing of the parameters and the way that they populate the space vary. In the particle filter, the basis is to represent the belief by a set of random state samples selected from this belief posterior and Figure 2.12 shows this basis.

In particle filtering, particles are used to represent the samples of posterior distribution:

$$\chi_t \coloneqq x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[N]} \tag{2.5}$$

Each particle $x_t^{[n]}$ (with $1 \leq n \leq N$) is used to instantiate the state at time t. N denotes the number of particles and χ_t denotes the particle set.

The insight behind the particle filters is that the belief $bel(x_t)$ is approximated by the particle set χ_t. Ideally, the likelihood of the state hypothesis shall be proportional to the belief:

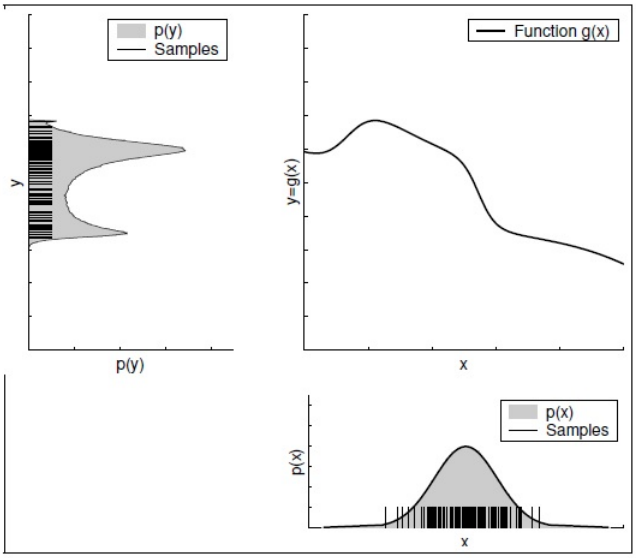


Figure 2.12: In the lower right, a graph is shown in which samples are drawn from Gaussian random variable, X. These samples are passed through the nonlinear function shown in the upper right graph. In the upper left, the resulting samples are distribution according to the random variable Y is shown [32].

$$bel(x_t^{[n]}) \backsim p(x_t|z_{1:t}, u_{1:t}) \tag{2.6}$$

Equation (2.6) results that the denser a sub-region of the state space is populated by samples, the more likely the true state falls into this region. The property (2.6) holds only $N \uparrow \infty$ for the standard particle algorithm. Although particles are drawn from a slightly different distribution for finite N, the effect of it can be neglected, if N is not too small.

Particle filter is a recursive algorithm. Hence, the current belief $bel(x_t)$ is calculated using the previous belief $bel(x_{t-1})$. Because the belief is denoted by the set of particles, it results that the current particle set χ_t is calculated using the particle set from previous step χ_{t-1}.

Table 2.2 shows the most basic variant of the particle filter. The algorithm has three inputs: particle set from the previous step χ_t, the most recent input u_t and the most recent measurement z_t. The algorithm firstly calculates the prediction function $\overline{bel}(x_t)$ to construct the particle set χ_t. It is applied for all the particles $x_{t-1}^{[n]}$ in the particle set

Table2.2: The most basic variant of particle filter based on importance sampling [32]

1:	**Algorithm Particle Filter**(χ_{t-1}, u_t, z_t)
2:	$\bar{\chi}_t = \chi_t = \varnothing$
3:	$for\ n = 1\ to\ N\ do$
4:	$sample\ x_t^{[n]} \backsim p(x_t \mid u_t, x_{t-1}^{[n]})$
5:	$w_t^{[n]} = p(z_t \mid x_t^{[n]})$
6:	$\bar{\chi}_t = \bar{\chi}_t + \langle x_t^{[n]}, w_t^{[n]} \rangle$
7:	$end\ for$
8:	$for\ n = 1\ to\ N\ do$
9:	$draw\ i\ with\ probability \propto w_t^{[i]}$
10:	$add\ x_t^{[i]}\ to\ \chi_t$
11:	$end\ for$
12:	$return\ \chi_t$

χ_t. The algorithm is described below in a detailed way:

1. In Line 4, by using the control input u_t and the particle $x_{t-1}^{[n]}$, a new particle representing the hypothetical state is generated. This step depends on the sampling from the state transition distribution $p(x_t|u_t, x_{t-1})$. After sampling the distribution, new particles which are the representations of $\overline{bel}(x_t)$ are obtained.

2. In Line 5, $importance factor$, denoted as w_t for each particle x_t, are calculated. That part of the algorithm actually means incorporating the measurement to the particle filtering process as $weight$ of the particle. The weight in that step represents the belief $bel(x_t)$.

3. $Resampling$ or $importance\ sampling$ is an important concept in particle filtering. In the resampling part of the algorithm, new N sized particle set is generated using the weights of the particles. Using weights means that particles with higher weight value will be most likely used for the new set χ_t, while particles with lower weight value will be removed from the set χ_t. Before the resampling part, particles are distributed with respect to $\overline{bel}(x_t)$, while after the resampling part they will be distributed with respect to the posterior $bel(x_t) = \eta\ p(z_t|x_t^{[n]})\overline{bel}(x_t)$. Furthermore, the resampling method in our implementation is explained in Section 2.5.4.3.

As it can be seen, the particle filter algorithm is compliant with the generic Bayes

algorithm shown in Table 2.1. In the next section, some properties of particle filter are explained and practical considerations about it are discussed.

2.5.4 Practical Considerations and Properties of Particle Filter

1) Density Extraction

The sample sets or particles employed by particle filters depict discrete approximations of continuous beliefs. However, many applications necessitate the availability of continuous estimates. In these applications, the estimates are not only the states represented by particles, but at any point in the state space. The deriving a continuous density from discrete samples is called $density\ estimation$ [32].

There are different ways for the density extraction and Fig. 2.13 depicts these ways. Figure 2.13(a) shows the particles and the density of the transformed Gaussian from the standard example given in Figure 2.12. As it is depicted with dashed lines in Figure 2.13(b), $Gaussian\ approximation$ is a simple and highly efficient technique to extract a density from particles. In this approach, the Gaussian extracted from particles is identical to the Gaussian approximation of the true density that is shown with solid line in Figure 2.13(b).

It is obvious that with the Gaussian approximation only basic properties of a density can be captured for unimodal densities. As it is shown in Figure 2.13(b), the function model is not well extracted from particles. Hence, more complex techniques, such as $k-means\ clustering$, are required. In that clustering technique, mixtures of Gaussians are used to approximate a density.

An alternative approach to the Gaussian is depicted in Figure 2.13(c). In this method, $histogram$ approach is used and a discrete histogram is constituted using bins distributed over the state space. In order to compute probability of each bin, weights of the particles whose values is located in the range of the bin is summed. This method has an important disadvantage which is with increasing number of bins, the space complexity is increasing exponentially. Nonetheless, that method has three main advantages. Firstly, histograms can illustrate multi-modal distributions. Secondly, they can be computed efficiently.

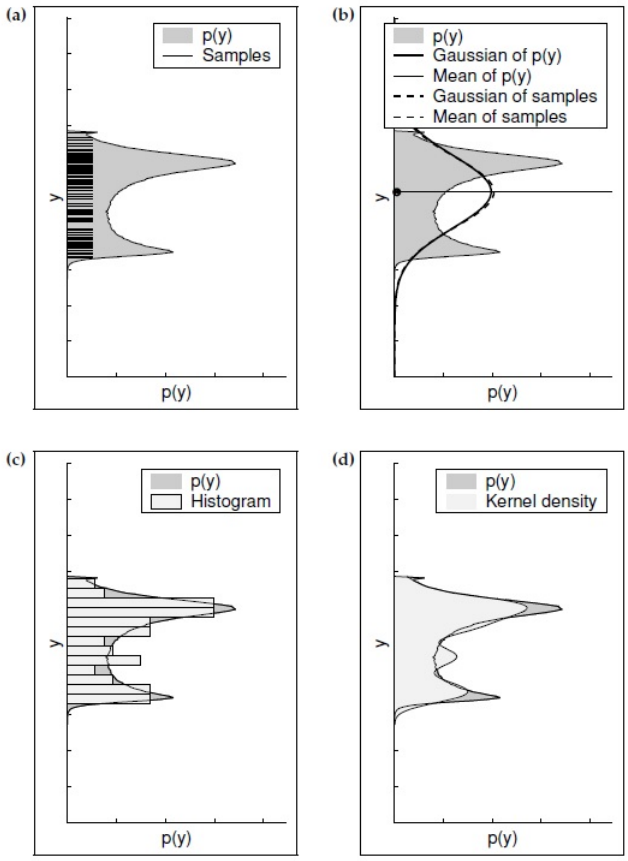


Figure 2.13: Different ways of extracting densities from particles. (a) Density and sample set approximation, (b) Gaussian approximation(mean and variance), (c) histogram approximation, (d) kernel density estimate. The choice of approximation strongly depends on the specific application and the computational resources [32].

Lastly, the density at any state can be derived from in time and it is independent of the number of particles.

A particle set can be converted into a continuous density with $kernel\ density\ estimation$. In this method, each particle in the set is used as the center of a kernel and the overall density is given by the mixture of the kernel densities. Figure 2.13(d) illustrates the result of the kernel method. The advantages of the method lies in the smoothness of the density estimate and the algorithmic simplicity. Yet, the complexity of computing the density is linearly related with the number of particles or number of kernels.

Choice of the method depends on the application area. Since, computational

power and estimation accuracy of the system directly affect the type of the model that will be implemented.

2) Sampling Variance

The variation which is inherent in random sampling is the main source of the error in the particle filter. It is inevitable that the representation of the original density with a finite number of samples differs a little from the original density. As an illustration, if a Gaussian distribution is modeled with a finite number of particles or samples, then the mean and the variance of the samples will be different from the original density. The variability caused by random sampling is named as the $variance$ of the sampler.

By considering two robots performing identically with the same noise-free actions. Obviously, it is expected that both robots should have same belief after performing the action. Simulation results of this situation is shown in Figure 2.14. Samples are drawn repeatedly from a Gaussian density and passed through a nonlinear transformation for the simulation. The resulting samples and their kernel density estimates are shown with the true belief depicted as gray area. In the graphs shown in the left part, results are prepared with 25 samples from the Gaussian, while the right part shows the results with 250 samples. As it can be concluded from these graphs, with increasing number of samples, the sampling variance between true belief and estimation is decreased and the observations made by the robot will be close enough to the true belief.

3) Resampling

Applying repetitive resampling causes amplification of the sampling variance. By considering an extreme case with a robot whose state does not change, source of this amplification can be better understood. Also, this situation can be described as $x_t = x_{t-1}$. By assuming that the robot has no sensors, the state of the robot can not be estimated. Apparently, that robot can not determine anything about its location, at any time instance t. Hence, its estimate at time t will be identical to its initial estimate.

However, if a vanilla particle filter is used, that won't be the case. After the initialization, generated particles will be dispersed throughout the state space.

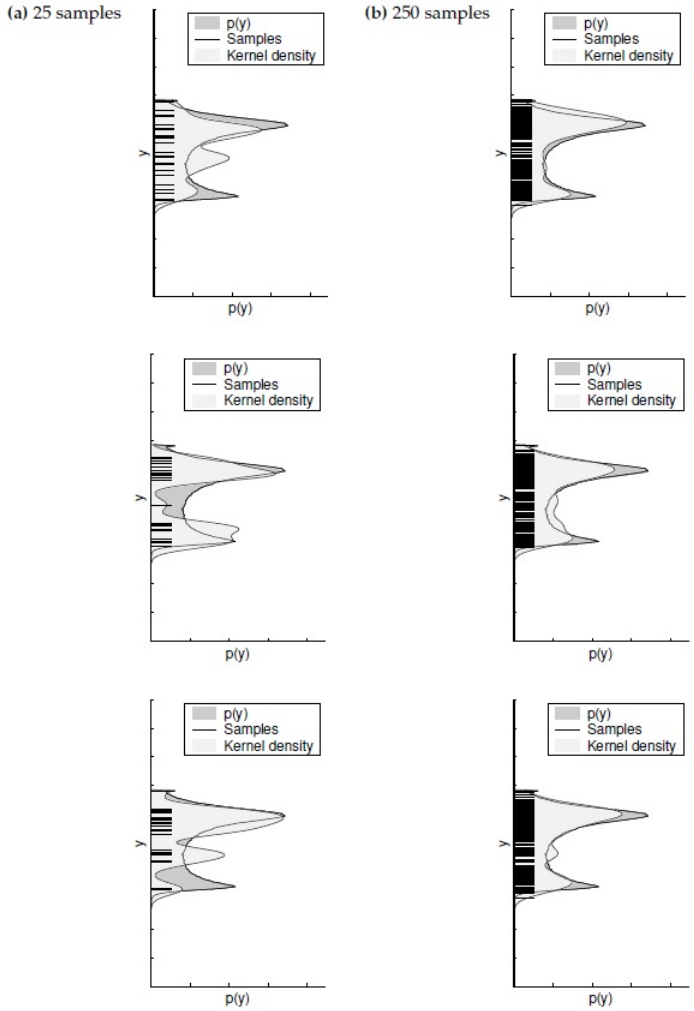


Figure 2.14: Variance due to sampling. Samples are drawn from a Gaussian and passed through a nonlinear function. Samples and kernel estimates resulting from repeated sampling of 25 (left column) and 250 (right column) samples are shown. Each row shows one random experiment. [32].

Nevertheless, because the state transition is deterministic, the fail of the resampling steps shown between line 8 and line 11 in Table 2.2 for reproducing a state sample $x^{[n]}$ is inevitable. Hence, no new states will be observed in the forward sampling state (line 4 of Table 2.2). As the iterations goes on, different particles will be deleted from the particle set, while no creation of the particles is observed. Consequently, N identical copies of a single particle will survive and in other words, the diversity will disappear. It seems to be the robot has determined its state with respect to an outside observer. However, that situation

is an obvious contradiction for a robot with no sensors on it.

This example also indicates another disadvantage of the particle filters with important practical results. Specifically, due to the resampling process the diversity in the particle population is decreased. Actually, this loss demonstrates itself as an approximation error. In the resampling, the sampling variance of the particle set itself decreases. However, as an estimator of the true belief, the variance, of the particle set increases. For any practical implementation, controlling of this variance or error is important.

For $variance\ reduction$, there exists two major strategies. First strategy is the reduction of the resampling frequency. In this technique, no resampling takes place, if the state is static or $x_t = x_{t-1}$. If the robots stops, then the resampling suspends. Even if the state changes, reduction of the resampling is a good idea. Integration of the multiple measurements can be always integrated via multiplicatively updating the importance factor. That idea can be illustrated as below:

$$w_t^{[n]} = \begin{cases} 1 & \text{if resampling took place.} \\ p(z_t|x_t^{[n]})w_{t-1}^{[n]} & \text{if no resampling took place.} \end{cases} \tag{2.7}$$

The choice of when to resample is complicated and necessitate practical experience. Because if the resampling is executed too often, then the risk of losing diversity increases. However, too infrequent resampling causes many samples to be wasted in regions of low probability. The measurement of the variance of the importance weights is a standard approach to deciding whether or not resampling should be performed.

The second strategy for the reduction of the sampling error is $low\ variance\ resampling$ and Table 2.3 illustrates the algorithm of it. In this algorithm, samples are not selected independently of each other in the resampling process for the basic filter given in Table 2.2. Instead of that type of selection, sequential stochastic process is chosen. The algorithm in Table 2.3 computes only one random number and selects samples according to this number. Yet, that number is selected with a proportional probability to the sample weight. That number is generated by drawing a random number r in the interval [0;N^{-1}]. The al-

Table2.3: Low variance resampling for the particle filter [32]

1:	**Algorithm Low Variance Sampler**(χ_t, W_t)
2:	$\bar{\chi}_t = \varnothing$
3:	$r = rand(0; N^{-1})$
4:	$c = w_t^{[1]}$
5:	$i = 1$
6:	$for\ n = 1\ to\ N\ do$
7:	$U = r + (n-1) \cdot N^{-1}$
8:	$while\ U > c$
9:	$i = i + 1$
10:	$c = c + w_t^{[i]}$
11:	$end\ while$
12:	$add\ x_t^{[i]}\ to\ \bar{\chi}_t$
13:	$end\ for$
14:	$return\ \bar{\chi}_t$

gorithm continue to work by adding N^{-1} to random number r repeatedly and the particle corresponding to that value is selected. U values in [0;1] indicates specifically one particle i:

$$i = \underset{j}{\operatorname{argmin}} \sum_{n=1}^{j} w_t^{[n]} \geq U \tag{2.8}$$

There are two reasons to use while loop in the algorithm of low variance sampler. Firstly, it computes the sum in the right hand side of this equation. Secondly, it checks whether i is the index of the first particle such that the corresponding sum of weights exceeds U. Line 12 shows the execution of the selection. Furthermore, Figure 2.15 depicts this process.

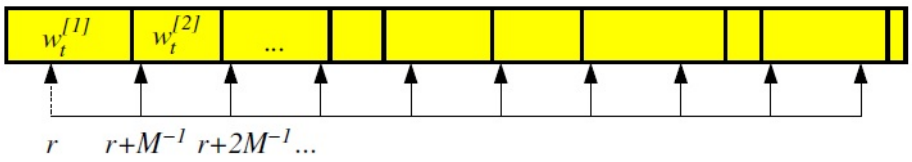


Figure 2.15: Working basis of the low variance resampling method. A random number r is chosen and then those particles corresponds to $u = r + (n-1) \cdot N^{-1}$ where $n = 1, ..., N$ [32].

The low-variance sampler has three main advantages. First, it covers the space of samples in a more systematic way than the independent random sampler. Since, the dependent sampler cycles through all particles systematically. Second, if all the samples have the same weight, then all of the samples are represented. Third, complexity of low variance sampler is $O(N)$. Although, achieving this complexity for an independent sampling method is difficult. This dependent sampler method has $O(NlogN)$ complexity.

In the implemented particle filter methods in this thesis, sampling importance resampling algorithm given in Table 2.3 is implemented in the resampling part of the particle filter.

4) Sampling Bias

Only finitely many particles are used in the particle filter and that introduces a systematic bias in the posterior estimate. For the extreme case of $N = 1$, the loop in lines 3 through 7 in Table 2.2 will be executed only once. The main observation is that the resampling step(from Line 8 to Line 11 of Table 2.2) deterministically accept this sample, regardless of the its weight $w_t^{[n]}$. Thus, measurement plays no role in the update. Particles are generated from $p(x_t|u_{1:t})$ instead of $p(x_t|u_{1:t}, z_{1:t})$. In other words, it ignores all measurements.

The normalization is the reason for the situation. Also, this normalization is implicit in the resampling step. When sampling in proportion to the importance weights (Line 9 of Table 2.2) $w_t^{[n]}$ becomes its own normalizer if $N = 1$:

$$p(\text{draw } x_t^{[n]} \text{ in line 9}) = \frac{w_t^{[n]}}{w_t^{[n]}} = 1 \tag{2.9}$$

In general, the problem is that although after normalization the non-normalized values $w_t^{[n]}$ reside in a space of dimension $N - 1$, they are drawn from an N-dimensional space. The reason for that is the n-th weight can be recovered from the $N - 1$ other weights by subtracting those from 1 after normalization. The effect of loss of dimensionality or degrees of freedom, becomes less observable for larger values of N.

5) Particle Deprivation

The problem that there are no particles in the vicinity of the correct state is known as the *particle deprivation problem*. Although the main reason for that situation is the small number of particles, that deprivation can be observed in any particle set size.

Another reason for that deprivation is the variation in random sampling. In some situations, unfortunately, the true states of the particles can be wiped out and the particles with incorrect states are generated.

A popular approach to solve this problem is to add some random particles to particle set χ_t after the resampling process, regardless of the actual sequence of motion and measurement commands. This approach is preferred due to its simplicity in the application.

A random particle injection method is presented in Thrun $et\ al.$ [32] and that method is also implemented in last method of the implementations in this thesis. Pseudo-code of the algorithm is given in Table 2.4.

The given random particle injection algorithm is adaptive and tracks the short term and the long-term average of the likelihood $p(z_t|z_{1:t-1}, u_{1:t}, n)$. The first part of the algorithm contains the motion model and the measurement model. It is similar to the basic particle filter algorithm given in Table 2.2. New poses are sampled from old particles in Line 5 and their importance weight is set with respect to the measurement model given in Line 6. In Line 8, the empirical measurement likelihood is calculated. It maintains short-term and long-term averages of this likelihood in lines 10 and 11. The values of α_{slow} and α_{fast} should be chosen as $0 \leq \alpha_{slow} \ll \alpha_{fast}$. α_{slow} is the parameter to represent the decay rate for exponential filter that estimates the long-term average, while α_{fast} is used for the short-term average. The critical part of the algorithm starts with Line 13. Probability of $max\{0.0, 1.0 - w_{fast}/w_{slow}\}$ is used to generate a random particle during the resampling. If a new particle is generated, the chosen resampling methods is executed. The divergence between the short-term and the long-term averages of the measurement likelihood is analyzed to determine the probability of adding a random sample. The chosen resampling method is applied, if the short-term likelihood is better or equal to the

Table2.4: Algorithm of the random particle injection [32].

1: **Random Particle Injection Algorithm**(χ_t, u_t, z_t, n)

$static\ w_{slow}, w_{fast}$

$\bar{\chi}_t = \chi_t = \varnothing$

$w_{avg} = 0;$

$for\ n = 1\ to\ N\ do$

$x_t^{[n]} = \mathbf{sample\_motion\_model}(u_t, x_{t-1}^{[n]})$

$w_t^{[n]} = \mathbf{measurement\_model}(z_t, x_t^{[n]}, n)$

$\bar{\chi}_t = \bar{\chi}_t + < x_t^{[n]}, w_t^{[n]} >$

$w_{avg} = w_{avg} + \frac{1}{N} w_t^{[n]}$

$end\ for$

$w_{slow} = w_{slow} + \alpha_{slow}(w_{avg} - w_{slow})$

$w_{fast} = w_{fast} + \alpha_{fast}(w_{avg} - w_{fast})$

$for\ n = 1\ to\ N\ do$

$with\ probability\ max\{0.0,\ 1.0 - w_{fast}/w_{slow}\}\ do$

$add\ random\ pose\ to\ \chi_t$

$else$

$draw\ i \in \{1, \ldots, N\}\ with\ probability \propto w_t^{[i]}$

$add\ x_t^{[i]}\ to\ \chi_t$

$end\ with$

$end\ for$

$return\ \chi_t$

long-term likelihood. In other words, no random sample is added. Otherwise, random samples whose number is related to the quotient of short- and long-term values are added to particle set. As it is explained, an increased number of random samples are induced by an abrupt decay in the measurement. The risk of mistaking momentary sensor noise for a poor localization result is prevented by the exponential smoothing.

2.5.5 Advantages and Disadvantages of Particle Filtering

Using particle filtering as the tracking approach has several advantages and disadvantages. Main advantages can be listed as[19]:

- It is applicable to nonlinear systems.
- It works with non-Gaussian noise.
- It has an adaptive behavior focusing only on the probable regions of state space.
- It has the ability to represent arbitrary densities.
- Its framework allows for including multiple models for tracking maneuvering targets.

Using particle filtering also has some disadvantages:

- In particle filtering, it is difficult to determine the optimal number of particles.
- It has high computational complexity. That complexity depends on number of particles.
- The need for number of particles increases with increasing model dimension.
- It may degenerate and lose the diversity in some cases.
- It is critical to choose importance density.

Table2.5: Particle filter based audio-visual tracking techniques

Title of the work	Audio Features	Video Features	Publication and Year
"Sequential Monte Carlo fusion of sound and vision for speaker tracking"	TDOA	Gradient	Vermaak et al. [34], 2001
"Audio-visual speaker tracking with importance particle filters"	TDOA	Coordinates	Perez et al. [12], 2002
"Audio assisted robust visual tracking with adaptive particle filtering"	DOA	Pixels of video frame	Kılıç et al. [18], 2015
"Joint audio-visual tracking using particle filters"	TDOA	Skin color, shape matching and color histogram	Zotkin et al. [38], 2002
"A joint particle filter for audio-visual speaker tracking"	TDOA	Haar-like features	Nickel et al. [27], 2005
"Audiovisual probabilistic tracking of multiple speakers in meetings"	TDOA	Shape and spatial structure of human heads	Gatica-Perez et al. [13], 2007
"Multi-level particle filter fusion of features and cues for audio-visual person tracking"	TDOA	Color, upper body detection and person region cues	Bernardin et al. [5], 2008

2.5.6 Particle Filter Based Audio-Visual Human/Object Tracking Methods

There are many application areas of particle filter based audio-visual methods as it is mentioned in Section 2.5.1. The attention of the thesis concentrates on the human tracking problem. Different approaches for human or object tracking problem by using particle filter based audio-visual methods are presented in Table 2.5 from surveys of [17], [30] and [2].

Following Chapter describes the AV16.3 Dataset which is used for the simulations of the implemented methods.

CHAPTER 3

AV16.3 DATASET

3.1 Introduction

In this thesis, a multi-person tracking system in a noisy and reverberant indoor environment is implemented as it is stated previously. For the performance evaluation of the tracking system, AV16.3 dataset[23] is used. It is a publicly available dataset and can be downloaded from Idiap Research Institute website by signing a "End User License Agreement".

"16.3" in the name of the dataset stands for 16 microphones and 3 cameras. This dataset provides an evaluation database for audio and visual tracking algorithms in the meeting room context.

In the design of that corpus, two contradicting constraints are considered:

1. By considering both "meeting situations" and "motion situations", the occupied area by the speakers should be large enough.

2. Occupied area should be completely visible by all cameras.

3.2 Physical Setup

The possible speakers' locations are selected as gray L-shaped area and shown in Figure 3.1. In this figure, three cameras are indicated by C1, C2 and C3, while two microphone arrays are indicated by MA1 and MA2. The field of view of all three

cameras are shown as gray area in the figure. This L-shaped area is a 3 m-long by 2 m-wide rectangle, minus a 0.6 m-wide potion taken by the tables.

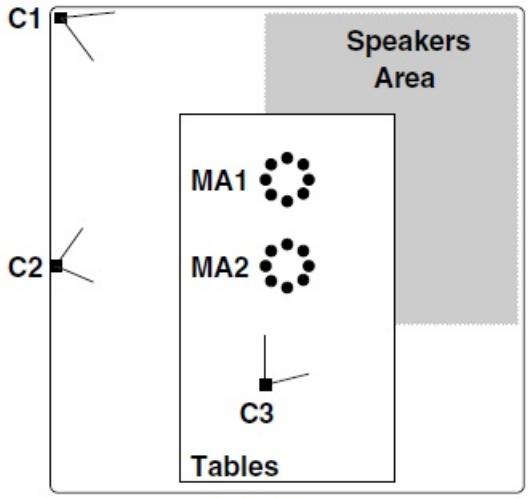


Figure 3.1: Physical setup of AV16.3 Dataset [23].

3.3 Hardware

In order to prepare this corpus, 3 cameras and 2 microphone arrays are used as stated previously and the instrumented room [26] is used. The resolution of each camera is 288x360 pixels and fps(frame per second) value is 25 Hz. Each microphone array is composed of 8 elements placed in a circular manner with 10 cm radius. The distance between center of the microphone arrays is 0.8 m.

3.4 Online Corpus

Each recorded sequence contains:

- 3 video files whose format is DIVX AVI and the resolution 288x360 for each camera. They are sampled at 25 Hz. Also, each video sequence contains one audio signal.

- 16 audio files whose format is WAV and recorded from two circular 8-microphone arrays sampled at 16 kHz.
- Additionally, for some sequences, more audio WAV files recorded from lapels worn by the speakers and sampled at 16 kHz. These audio files are not used in the thesis.

For localization and tracking purposes, cases listed below are included in sequences:

- Overlapped speech
- Close and far locations, small and large angular separations
- Object initialization
- Variable number of objects
- Partial and total occlusions
- Natural changes of illumination

For any sequence, there are at most three people. In these sequences, motion of the speaker are static (e.g. seated person) or dynamic (e.g. walking person) or could be mixed.

3.5 Content

Brief explanations about the annotated sequences with videos given below.

seq01-1p-0000 One speaker is static while speaking. The speaker stands at each of 16 different locations in the shaded area shown in Figure 3.1. During the speech, the speaker faces the microphone arrays. The aim of this sequence is to evaluate the audio localization for the single speaker case.

seq11-1p-0100 A single speaker mostly moves during speaking and makes abrupt moves. However, the speaker faces the microphone arrays. The purpose of this sequence is to test audio, video and audio-visual speaker cases specifically for difficult motion scenarios.

seq15-1p-0100 In this one moving speaker case, there are alternating speech and long silences while the speaker is walking. The aim of this sequence is to:

1. Demonstrate that the audio tracking is not capable to recover from unpredictable trajectories during silence,
2. Present an initial test case for AV tracking.

seq18-2p-0101 There are two speakers in the sequence. They talks and faces the microphone arrays during the sequence. They slowly gets as close as possible to each other, then slowly parting. Multi-source localization, tracking and separation are motivation of the sequence.

seq24-2p-0111 In this case with two moving speakers, they cross the field of view twice each other and also occludes each other twice. Additionally, they talks most of the time. The purpose of the sequence is to test both audio and video occlusions.

seq40-3p-0111 There are three speakers in this sequence. Two of them are seated and one standing. During the sequence all of them speaks continuously and faces the arrays. Also, the standing speaker walks back and forth once behind the seated speakers. The purpose is both to test multi-source localization, tracking and separation algorithms. Furthermore, this sequence aims to highlight complementarity between audio and video modalities.

seq45-3p-1111 In this sequence with three speakers, the motion of them is unconstrained. They enters and leaves the scene while they are speaking continuously and occluding each other many times. In this very difficult case, there are difficult case of overlapped speech and visual occlusions. The motivation is to highlight the complementarity between audio and video modalities.

In the simulation part, all of the annotated cases listed above are used. Ground truths with respect to 2-D pixel planes of each cameras and 3-D real world are provided by the dataset. In addition to the ground truths of the listed sequences, more ground truths for other sequences can be prepared using the tools provided by AV16.3. However, in this thesis, only the ground truths provided by the dataset is used. Table 3.1 summarizes what is explained about these annotated video sequences and presents

the duration of them. Further information is presented in [23]. Moreover, snapshots from different sequences are shown in Figure 3.2.

Table3.1: List of annotated video sequences[23] of AV16.3 Dataset. Meaning of tags: [A]udio, [V]ideo, predominant [(ov)]erlapped speech, at least one visual [(occ)]lusion, [S]tatic speakers, [D]ynamic speakers, [U]nconstrained motion.

Sequence Name	**Duration (seconds)**	**Modalities of Interest**	**Number of Speakers**	**Speaker(s) Behavior**
seq01-1p-0000	217	A	1	S
seq11-1p-0100	30	A, V, AV	1	D
seq15-1p-0100	35	AV	1	S, D(U)
seq18-2p-0101	56	A(ov)	2	S, D
seq24-2p-0111	48	A(ov), V(occ)	2	D
seq40-3p-0111	50	A(ov), AV	3	S, D
seq45-3p-1111	43	A(ov), V(occ), AV	3	D(U)

3.6 Camera Calibration

By assuming the middle point of first microphone array(MA1) and second microphone array (MA1 and MA2 as it can be seen in Figure 3.1) as the center, all of the calibration parameters and projection matrices explained in Section 2.3.2 are provided by AV16.3 dataset.

Following section explains the details of the eight implemented methods using the AV16.3 dataset that is described in this section.

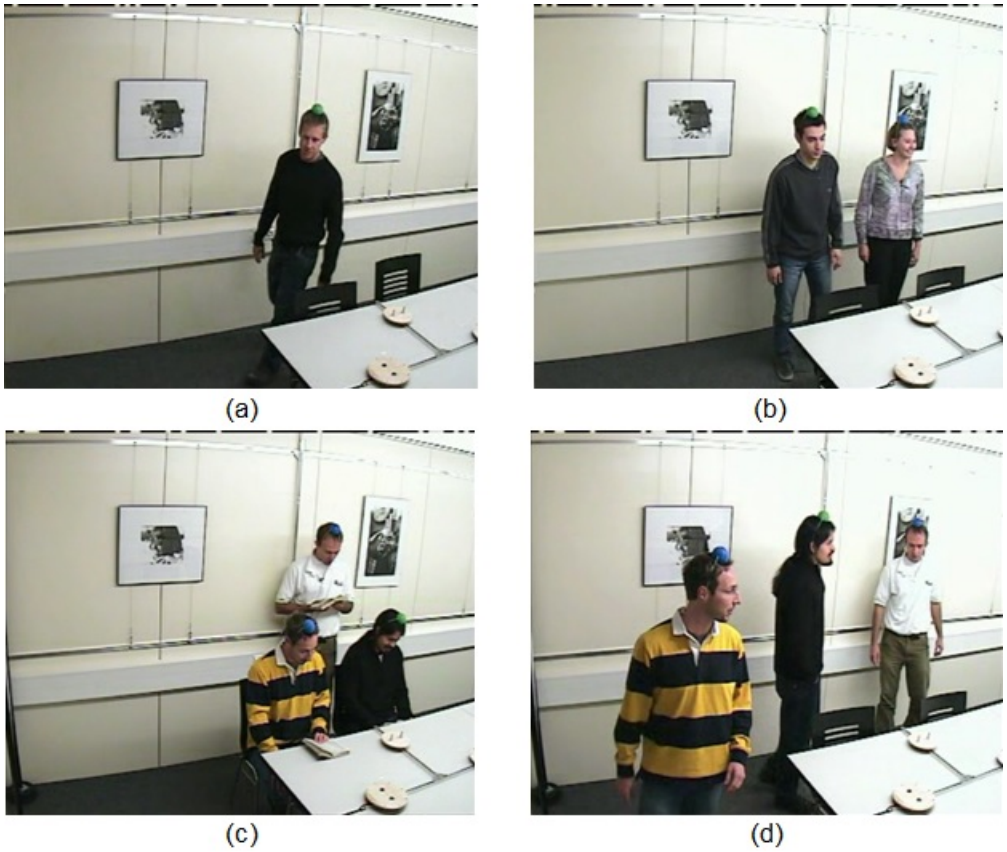


Figure 3.2: Snapshots from AV16.3 Dataset. (a)seq11-1p-0100-cam1, (b)seq18-2p-0101-cam1, (c)seq40-3p-0111-cam1, (d)seq45-3p-1111-cam1[23].

CHAPTER 4

IMPLEMENTATION OF THE TRACKING METHODS FOR MULTIPLE SPEAKER TRACKING

4.1 Introduction

All the implemented methods in this thesis aims to track the speaker or speakers. For this purpose, except two methods(V-PF and V-PF-2CAM), audio and visual cues are integrated to locate the speaker's position. Also, these two methods are implemented to analyze the contribution of the audio integration to the visual-only tracker.

For the visual part, Bhattacharyya distance is used as the distance measure and implementation details are given Section 4.2. Although the method proposed by Kılıç $et\ al.$[18] is used in this thesis, there are similar and successful implementations of Bhattacharrya distance in particle filtering framework. Pérez $et\ al.$[29] propose a method for multiple face tracking with manual and automatic initialization using Bhattacharyya distance and particle filtering. Brassnet $et\ al.$[7] suggest a technique for object tracking in video sequences using particle filtering. Color, edge and texture cues are used for the tracking. For all of the cues, Bhattacharyya distance used as distance measure in histograms of the cues. In addition to these techniques, Nummiaro $et\ al.$[28] implement a robust color-based particle filtering method that also uses Bhattacharyya distance. In this work, after the implementation of the tracker, results of mean-shift and Kalman filtering trackers are compared with the particle filtering approach.

For the audio part, SSM method is implemented for all the methods uses the audio-

visual fusion. Implementation details are explained in Section 2.2 and Section 4.3.

After conducting a literature survey, firstly, recent V-PF and AV-PF methods in [18] are implemented for only the comparison. After that implementation, six different methods are implemented in order to construct a tracking system capable of 3-D tracking of multi-person with partial occlusion handling using 2 cameras and 2 microphone arrays. Implemented methods and their abbreviations are listed:

1. Visual particle filter(V-PF) technique from [18]
2. Audio-visual particle filter(AV-PF) technique from [18]
3. Particle filter based audio-visual tracking technique in 2-D(AV-PF-2D)
4. Particle filter based audio-visual tracking technique in 2-D with speech/non-speech classification(AV-PF-2D-SNS)
5. Particle filter based audio-visual technique in 3-D (AV-PF-1CAM-3D)
6. Particle filter based visual tracking technique in 3-D by using two cameras(V-PF-2CAM)
7. Particle filter based audio-visual tracking technique using two cameras and two microphone arrays(AV-PF-3D)
8. Particle filter based audio-visual fusion tracking technique in 3-D by using two cameras and two microphone arrays with occlusion handling (AV-PF-RAND)

Implementation details of these methods with their pseudo-codes are explained in the following sections.

4.2 Particle Filtering-Based Visual Tracking Method(V-PF)

In this particle filter-based tracking method, only one camera is used and the speaker is tracked in the image plane of the camera. This visual-only tracker method, implemented by Kılıç $et\ al.$ [18], is a recent method and builds a base to compare visual-only and audio-visual methods in 2-D. That V-PF method works in five steps. It uses

histogram based particle cue in order to weight particles. In the first step, the particles are initialized. $w_0^{(n)} = \frac{1}{N}$ for $n = 1, ..., N$. Here N is the number of particles and $w_0^{(n)}$ are the initial weights of the particles. In this technique, manual start approach is used and hence the initial particles are manually injected near the faces. In the particle filter, state vector is defined as $\mathbf{x} = [\ x_1\ \dot{x}_1\ x_2\ \dot{x}_2\ s\ ]^T$. In this definition, x_1 and x_2 are the horizontal and vertical positions of the rectangle centered around the face which is wished to track. $\dot{x}_1$ and $\dot{x}_2$ are horizontal and vertical velocities of the particle. Additionally, s is the scale of the rectangle centered around (x_1, x_2).

In the second step, the dynamic model given below is used for the propagation of the particle.

$$\mathbf{x}_k^{(n)} = \mathbf{F}\mathbf{x}_{k-1}^{(n)} + \mathbf{q}_k^{(n)} \tag{4.1}$$

where $\mathbf{x}_k^{(n)}$ is the state of the n-th particle at the frame $k = 1, ..., K$ and $\mathbf{q}_k^{(n)}$ is the zero-mean Gaussian noise with covariance $\mathbf{Q}$, $\mathbf{q}_k^{(n)} \backsim N(0, Q)$ for each particle and $\mathbf{F}$ is the linear motion model.

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} \sigma^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_s^2 \end{bmatrix}$$

In these matrices, T is the period between two adjacent frames. σ_s^2 is the variance of the scale and σ^2 is the variance for both the position and the velocity. This motion model assumes the same variance on both the position and the velocity. Also, the described linear motion model can be updated so that it fits realistic scenarios more. However, for V-PF and AV-PF are implemented with model[18] explained above. Last six methods are implemented using the new proposed motion model whose details are given in the related sections about the tracking techniques.

In the third step, particles are weighted with respect to observation model

$$w_k^{(n)} = e^{-\lambda(D^{(n)})^2} \tag{4.2}$$

In this weighting equation, λ is the design parameter and $D^{(n)}$ is the Bhattacharyya

Table4.1: Visual particle filter(V-PF) tracking algorithm [18] .

1:	Initialize $N, \sigma^2, U, T, \mathbf{F}, \lambda, r(u), \mathbf{x}_0^{(n)}, w_0^{(n)}, k$
2:	**while** $k < K$ **do**
3:	Propagate particles: $\mathbf{x}_k^{(n)} = \mathbf{F}\mathbf{x}_{k-1}^{(n)} + q_k^{(n)}$
4:	Calculate $D^{(n)}$ using Equation 4.3, for $n = 1, \ldots, N$
5:	Weighting: $w_k^{(n)} = e^{-\lambda(D^{(n)})^2}$, for $n = 1, ..., N$
6:	Normalization: Re-weight particles to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$
7:	Estimate target position $\tilde{\mathbf{x}}_k = \sum_{n=1}^{N} w_k^{(n)} \mathbf{x}_k^{(n)}$
8:	Resampling: Generate $\mathbf{x}_k^{(n)}$ from the set $\{\mathbf{x}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}$
9:	$k = k + 1$
10:	**end while**

distance

$$D^{(n)} = \sqrt{1 - \sum_{u=1}^{U} \sqrt{r(u)q^{(n)}(u)}} \tag{4.3}$$

In this equation, U is the number of the histogram bins, $r(u)$ is the Hue histogram of the reference image. Reference image of the target person is determined from the frames of the related sequence. Also, $q^{(n)}(u)$ is the Hue histogram extracted from the rectangle centered on the position of the n-th particle. HSV is preferred in this work, since it is more robust in variations of the illumination.

Lastly, in the third step, normalization is applied to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$.

In the fourth step, position of the speaker is estimated by

$$\tilde{x}_k = \sum_{n=1}^{N} w_k^{(n)} x_k^{(n)} \tag{4.4}$$

In the last step, the particles $x_k^{(n)}$ are resampled. The resampling process of low variance sampler explained in "Resampling" part of Section 2.5.4.4 is implemented. After that step, the algorithm continues to run recursively.

The algorithm explained above is shown as pseudo code in Table 4.1.

In addition to the algorithm given Table 4.1, one modification is applied to the algorithm. Horizontal and vertical velocities of the particle $\dot{x}_1$ and $\dot{x}_2$ are limited in terms of magnitude. Otherwise, high velocity of the particles may cause the parti-

cle deprivation explained in Section 2.5.4.5. Hence, the similar approaches about the limitation are applied for the remaining methods.

4.3 Particle Filter-Based Audio Constraint Visual Tracking Method(AV-PF)

In this algorithm, audio information is used to enhance the visual tracking described in Section 4.2. Also, the audio trackig method of Lathoud $et\ al.$ [22] which is summarized in Section 2.2 is implemented. The DOA(Direction of arrival) measurements of that algorithm are given in terms of azimuth angle and they are used in this method. Since the DOA measurements can be still noisy after the applying SSM method, a third order audio restoration model is applied to estimates in order to improve reliability of the azimuth.

$$\bar{\theta}_k = \sum_{i=0}^{2} \varphi_i \theta_{k-i} + \varepsilon_k \tag{4.5}$$

In this equation, θ_k is the azimuth angle estimate in terms of degrees and $\bar{\theta}_k$ is the filtered azimuth value. Additionally, φ_i is the parameter of the model to adjust rate of present value and past value for inclusion to the filter and ε_k is white noise for k-th image frame.

The geometric calibration is explained in Section 2.3.2. Geometric camera calibration parameters and locations of the microphones are provided by AV16.3 dataset [23]. Hence, all 3-D coordinates given in microphone coordinate system can be projected to image planes of the cameras.

The idea behind this method which aims to intagrate audio measurements into V-PF is given in [18]. In this method, the distributed particles are relocated around the DOA line. After that, the AV-PF re-calculate the weights of the relocated particles according to their distance to the DOA line. That DOA line can be drawn as follows. Firstly, 3-D position of the speaker's head (A, B_k, C) is determined based on the estimated DOA angle and following assumptions:

1. A is the distance from the center of the microphone array to the wall in meters and it is taken as 1.75 meters in implementations

2. C is the estimated height of the speaker, typically chosen as 1.80 meters in the implementation.

And, B_k is calculated using the trigonometric identity:

$$B_k = \tan(\theta_k \times \frac{\pi}{180}) \cdot A \tag{4.6}$$

After calculating the values of (A, B_k, C), these 3-D points are projected to the image frame to obtain the 2-D coordinates (a_k, b_k) using calibration matrix provided in the dataset. Lastly, the DOA line is drawn from (a_k, b_k) to the 2-D coordinate of the center of the first microphone array which shown as MA1 in Figure 3.1.

The main reason to draw the DOA line is to concentrate the particles around that line. Concentrating on the DOA line is likely to increase the possibility of the speaker detection, since that line indicates the approximate direction of the sound emanating from the speaker. Initially, the locations of the particles are distributed in a circular area. But, after that DOA line implementation, the distribution is elliptical. It is not exactly on the DOA line, since it is wished to avoid deviation in the detection regarding noisy DOAs measurements. To get the elliptical distribution, particles are moved toward to the DOA line with respect to the distance to it. In other words, farthest particle moves more than the closest particle. In order to achieve this movement, firstly the perpendicular distances to DOA line $\mathbf{d}_k = [d_k^{(1)} \ ... \ d_k^{(N)}]^T$ is calculated. After the calculation, they are normalized to obtain distance coefficients as follows:

$$\hat{\mathbf{d}}_k = \frac{d_k}{\|d_k\|} \odot \mathbf{d}_k \tag{4.7}$$

In this equation, $\odot$ is the element-wise product and $\|.\|_1$ is the l_1 norm. Also, the distance coefficients can be explained as $\hat{\mathbf{d}}_k = [\hat{d}_k^{(1)} \dots \hat{d}_k^{(N)}]^T$. These $\hat{\mathbf{d}}_k$ coefficients are used to calculate how much the particles will be moved towards the DOA line and it is be shown in the next step.

The noise within the audio measurements corrupts the reliability and accuracy of the DOAs. In order to prevent this corruption, integration of the audio measurement is controlled with Bhattacharyya distance calculated using Equation 4.3 for the target determined using only visual cue. Hence, the dynamic model given in Equation 4.1 is modified as:

$$\hat{\mathbf{x}}_k^{(n)} = \mathbf{x}_k^{(n)} \oplus \hat{d}_k^{(n)} \mathbf{h}_k \gamma_k \tag{4.8}$$

In this equation, $\oplus$ is the element-wise addition and $\mathbf{h}_k = [cos(\theta_k)\ 0\ sin(\theta_k)\ 0\ 0]^T$. In this equation, using $\hat{d}_k^{(n)}$, $\mathbf{h}_k$ and γ_k, only the position of the particles are updated. Also, a new method to calculate the importance weights is proposed as by including the audio measurement to Equation 4.2:

$$\hat{w}_k^{(n)} = (e^{-\lambda (D^{(n)})^2}) \frac{\|\mathbf{d}_k\|_1}{d_k^{(n)}} \tag{4.9}$$

After that, weights are normalized to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$. After weighting step, the position of the face is estimated using Equation 4.4 and denoted as $\tilde{x}_k^{av}$. Before the resampling step, to prevent the tracker to be falsified by the noise in the audio estimation, γ_k is calculated again with $\tilde{x}_k^{av}$ and denoted as γ_k^{av}. If γ_k^{av} is smaller than γ_k, the AV tracker results are used in the next step and iteration. Otherwise, audio is assumed to be noisy and the audio constrains on the estimation are ignored. Thus, visual-only tracker is used in the next step and iteration. In the last step, the resampling process sampling importance resampling is applied.

The algorithm of AV-PF explained above is given as pseudo code in Table 4.2.

With the proposed modifications in Equation 4.8 and Equation 4.9 with respect to visual-only V-PF tracker, the tracking algorithm preserve the position of the face even if the visual tracker is lost. Since concentrating particles around the DOA line increases the efficiency of the particles in terms of speaker detection, all particles converge the potential location of the speaker.

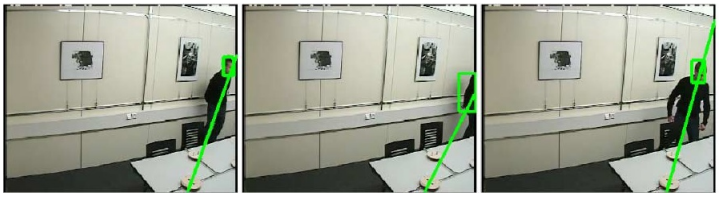


Figure 4.1: DOA lines in AV-PF method [18].

Table4.2: Audio-visual particle filter(AV-PF) tracking algorithm [18] .

Initialize $N, \sigma^2, U, T, \mathbf{F}, \lambda, r(u), \mathbf{x}_0^{(n)}, w_0^{(n)}, k$
while $k < K$ **do**
    Propagate particles: $\mathbf{x}_k^{(n)} = \mathbf{F}\mathbf{x}_{k-1}^{(n)} + q_k^{(n)}$
    Calculate $D^{(n)}$ using Equation 4.3
    Calculate weights: $w_k^{(n)} = e^{-\lambda(D^{(n)})^2}$, for $n = 1, ..., N$
    Normalization: Re-weight particles to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$
    Estimate target position $\tilde{\mathbf{x}}_k$ using Equation 4.4
    Calculate γ_k using Equation 4.3
    Get corresponding DOA angle θ_k
    Calculate distances $\mathbf{d}_k = [d_k^{(1)} \ ... \ d_k^{(N)}]^T$
    Find movement distances: $\hat{\mathbf{d}}_k = \frac{\mathbf{d}_k \odot \mathbf{d}_k}{\|\mathbf{d}_k\|_1}$
    Re-propagate particles: $\hat{\mathbf{x}}_k^{(n)} = \mathbf{x}_k^{(n)} \oplus \hat{d}_k^{(n)} \mathbf{h}_k \gamma_k$
    Re-weighting: $\hat{w}_k^{(n)} = (e^{-\lambda(D^{(n)})^2}) \frac{\|\mathbf{d}_k\|_1}{d_k^{(n)}}$
    Normalization: Re-weight particles to ensure that $\sum_{n=1}^{N} \hat{w}_k^{(n)} = 1$
    Re-estimate target position $\tilde{\mathbf{x}}_k^{av}$ using Equation 4.4
    Calculate γ_k^{av} using Equation 4.3
    if $\gamma_k^{av} < \gamma_k$ **then**
        $\mathbf{x}_k^{(n)} = \hat{\mathbf{x}}_k^{(n)}$, $w_k^{(n)} = \hat{w}_k^{(n)}$, $\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}_k^{av}$
    end if
    Resampling: Generate $\mathbf{x}_k^{(n)}$ from the set $\{\mathbf{x}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}$
    $k = k + 1$
end while

In the AV16.3 dataset, the speakers are talking continuously most of the time. Hence, using the DOA line has advantages to improve visual-only tracking. If there is no audio clue in the system, the DOA is estimated using last position of the DOA is available. But, if the gap between the two adjacent speaking frame is too large, then accuracy of the estimation will be limited. As a result of this, the target can be missed in these situations.

In this thesis, methods containing audio location estimation use annotated DOAs as a priori to avoid mis-correspondence of person-ID after occlusion. Actually, that information may not be available in a practical tracking system. Thus, the methods in [4] and [20] can be used for modeling of the person-IDs.

4.4 Particle Filter Based Audio-Visual Tracking Technique in 2-D(AV-PF-2D)

The motion and the sensor models are two basic parts of the particle filter as stated previously. Both models differ from previous two methods explained in Section 4.2 and 4.3. For the motion model, a new variance value is defined for the position and the velocity of the particles. But, a vector **G** is added to the motion model in order to reflect different noise on the position and the velocity. Additionally, by considering the speaker motion in scene, "Nearly Constant Velocity Model" described in Bar-Shalom $et\ al.$ [3] and Blair [6] is chosen to be implemented in the tracker systems.

In any dynamical system, equations of the motion are explained as:

$$x = x_0 + v * t + \frac{a * t^2}{2} \tag{4.10}$$

$$v = v_0 + a * t \tag{4.11}$$

In this equations, position, velocity and acceleration are denoted by x, v and a respectively. Thus, by considering these equations, Equation 4.1 is modified as:

$$\mathbf{x}_k^{(n)} = \mathbf{F}\mathbf{x}_{k-1}^{(n)} + \mathbf{G}\mathbf{q}_k^{(n)} \tag{4.12}$$

In this equation, **F** and **G** used to represent linear motion model:

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{G} = \begin{bmatrix} \frac{T^2}{2} \\ T \\ \frac{T^2}{2} \\ T \\ 1 \end{bmatrix} \tag{4.13}$$

$\mathbf{q_k^{(n)}}$ is the zero-mean Gaussian noise with covariance **Q** and $\mathbf{q}_k^{(n)} \backsim N(0, Q)$ for each particle as stated in previous methods. But, the content of the **Q** is redefined as:

$$\mathbf{Q} = \begin{bmatrix} \sigma^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_s^2 \end{bmatrix} \tag{4.14}$$

In this matrix, σ^2 is the variance of the position and velocity. Also, σ_s^2 is the variance for the scale.

In the remaining methods, except V-PF-2CAM, location estimates from video and audio are integrated. In 2-D tracking methods, audio data are collected from two microphone arrays and video data are collected from one camera. However, in 3-D tracking methods, two cameras are used with two microphone arrays. From these data, location of the speaker are estimated separately. In order to integrate these estimates, it is assumed that the measurements are extracted for each person and different measurements are conditionally independent for given states of the single person. These conditional independence explanation is presented in [29] and [13]. Hence, the measurements Z_t in the 3-D tracking techniques can be described at time t as:

$$Z_t = (z_t^{audio}, z_t^{video,cam1}, z_t^{video,cam2}) \tag{4.15}$$

Also, for a given state, the measurements produce following factorized representation:

$$p(Z_t|x_t) = p(z_t^{audio}|x_t) * p(z_t^{video,cam1}|x_t) * p(z_t^{video,cam2}|x_t) \tag{4.16}$$

Since the measurements are directly related with the weighting in the particle filtering context, Equation 4.16 means that the product of different measurements can be used for the calculation of the weight for a given state.

In the visual part remaining techniques, the method explained in Section 4.2 is used. For the audio part, contrary to AV-PF method explained in 4.3, two microphone arrays in AV16.3 dataset are used instead of one. Lathoud $et\ al.$ [21] indicates that only the estimates about azimuth angle of the proposed algorithm is reliable for a reverberant and noisy closed room. Hence, in order to increase the accuracy of the position estimation, geometrical crossings of the azimuth estimates is used for each frame. That crossing approach is shown in Figure 4.2. In this figure, azimuth angles with respect origins(o_1 and o_2) of two microphone arrays(MA1 and MA2) are shown as θ_1 and θ_2 respectively.

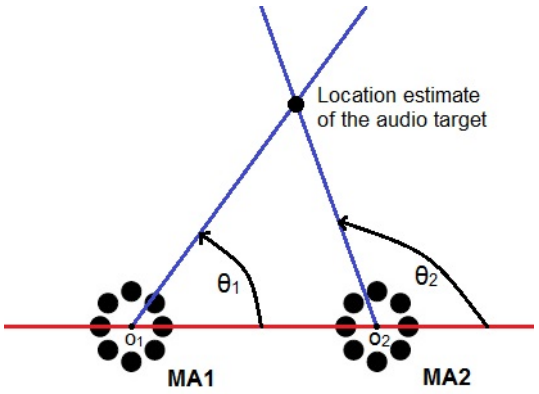


Figure 4.2: Location estimate of the audio source in x-y plane using two microphone array.

After estimating the position of the speaker on x-y axis, the position on the z-axis is generated considering the height of a normal person. For each iteration, a random number between 1.5 m and 1.8 m is generated and used as the height of the person. After estimating 3-D coordinates, that estimation is projected to the image plane using

the related projection matrix provided by AV16.3 dataset. In order to weight the particle using only the audio measurement, Euclidean distance between the related particle position and the projection is calculated as:

$$D_{audio}^{(n)} = \sqrt{(\check{x}_{particle,pos}^{(n)} - \check{x}_{audio,est})^2 + (\check{y}_{particle,pos}^{(n)} - \check{y}_{audio,est})^2} \tag{4.17}$$

The $\check{x}_{particle,pos}^{(n)}$ and $\check{y}_{particle,pos}^{(n)}$ are the position of the n-th particle in x and y axes on the image plane, while $\check{x}_{audio,est}$ and $\check{y}_{audio,est}$ are the estimated location of the audio source in x and y axes on the image plane respectively. The distance in Equation 4.17 is used for audio weighting as:

$$w_k^{(n)} = e^{-\lambda_{audio}(D_{audio}^{(n)})^2} \tag{4.18}$$

In this equation, λ_{audio} is the weighting parameter for audio measurements and used as an input to the algorithm.

In this 2-D tracker method, only one camera is used for the visual tracker. Hence, Equation 4.16 becomes:

$$p(Z_t|x_t) = p(z_t^{audio}|x_t) * p(z_t^{video}|x_t) \tag{4.19}$$

In the particle filtering framework, the weight of a particle is calculated using the measurement value as it is stated in Line 5 of the basic particle filter algorithm shown in Table 2.2. By combining Equation 4.19 and 4.2, the weight of the particle is calculated as:

$$w_k^{(n)} = e^{-\lambda(D^{(n)})^2} * e^{-\lambda_{audio}(D_{audio}^{(n)})^2} \tag{4.20}$$

Contrary to method AV-PF, no audio restoration model is used for the remaining six methods. In order to handle with the noise on the measurements, a pre-determined parameter(τ) is used. If the weight of the audio measurement is lower than this value, then the audio measurement is ignored. Aim of this method is to discard irrelevant au-

Table4.3: Algorithm of the particle filter based audio-visual tracking technique in 2-D(AV-PF-2D).

1:	Initialize $N, \sigma^2, U, T, \mathbf{F}, \lambda, \lambda_{audio}, \tau, r(u), \mathbf{x}_0^{(n)}, w_0^{(n)}, k$
2:	**while** $k < K$ **do**
3:	Propagate particles: $\mathbf{x}_k^{(n)} = \mathbf{F}\mathbf{x}_{k-1}^{(n)} + \mathbf{G}q_k^{(n)}$
4:	Calculate $D^{(n)}$ for visual weights using Equation 4.3
5:	Calculate visual weights: $w_{visual,k}^{(n)} = e^{-\lambda(D^{(n)})^2}$, for $n = 1, ..., N$
6:	For audio component, use the crossing point of the azimuth angles from two microphone arrays as it is described in Figure 4.2 in 3-D. Estimate the 3-D position of the mouth of the speaker by randomly generating the height between 1.5 m and 1.8 m and project the 3-D coordinate to the image plane.
7:	Calculate $D_{audio}^{(n)}$ for audio using Equation 4.17
8:	Calculate audio weights: $w_{audio,k}^{(n)} = e^{-\lambda_{audio}(D_{audio}^{(n)})^2}$, for $n = 1, ..., N$
9:	Average of the audio measurements: $w_{av,audio} = \frac{\sum_{i=1}^{N} w_{audio,k}^{(n)}}{N}$
10:	**if** $w_{av,audio} > \tau$ **then**
11:	$w_k^{(n)} = w_{audio,k}^{(n)} * w_{visual,k}^{(n)}$
12:	**else**
13:	$w_k^{(n)} = w_{visual,k}^{(n)}$
14:	**end if**
15:	Normalization: Re-weight particles to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$
16:	Estimate target position $\tilde{\mathbf{x}}_k = \sum_{n=1}^{N} w_k^{(n)} \mathbf{x}_k^{(n)}$
17:	Resampling: Generate $\mathbf{x}_k^{(n)}$ from the set $\{\mathbf{x}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}$
18:	$k = k + 1$
19:	**end while**

dio measurements. In addition to these, for nonspeaking intervals, location estimation from last available measurement is used.

AV-PF-2D algorithm explained in this section is given as pseudo code in Table 4.3.

4.5 Particle Filter Based Audio-Visual Tracking Technique in 2-D with Speech/Non-Speech Classification(AV-PF-2D-SNS)

For this method, AV-PF-2D technique described in Section 4.4 is modified by adding the speech/non-speech classification for the speaker. With the speech/non-speech classification, if the target speaker is silent, then the audio component for the weight-

Table4.4: Algorithm of the particle filter based audio-visual tracking technique in 2-D with speech/non-speech classification(AV-PF-2D-SNS).

Initialize $N, \sigma^2, U, T, \mathbf{F}, \lambda, \lambda_{audio}, \tau, r(u), \mathbf{x}_0^{(n)}, w_0^{(n)}, k$
while $k < K$ **do**
    Propagate particles: $\mathbf{x}_k^{(n)} = \mathbf{F}\mathbf{x}_{k-1}^{(n)} + \mathbf{G}q_k^{(n)}$
    Calculate $D^{(n)}$ for visual weights using Equation 4.3
    Calculate visual weights: $w_{visual,k}^{(n)} = e^{-\lambda(D^{(n)})^2}$, for $n = 1, ..., N$
    For audio component, use the crossing point of the azimuth angles from two microphone arrays as it is described in Figure 4.2 in 3-D. Estimate the 3-D position of the mouth of the speaker by randomly generating the height between 1.5 m and 1.8 m and project the 3-D coordinate to the image plane.
    Calculate $D_{audio}^{(n)}$ for audio using Equation 4.17
    Calculate audio weights: $w_{audio,k}^{(n)} = e^{-\lambda_{audio}(D_{audio}^{(n)})^2}$, for $n = 1, ..., N$
    Average of the audio measurements: $w_{av,audio} = \frac{\sum_{i=1}^{N} w_{audio,k}^{(n)}}{N}$
    if The speaker is not silent **then**
        if $w_{av,audio} > \tau$ **then**
            $w_k^{(n)} = w_{audio,k}^{(n)} * w_{visual,k}^{(n)}$
        else
            $w_k^{(n)} = w_{visual,k}^{(n)}$
        end if
    else
        $w_k^{(n)} = w_{visual,k}^{(n)}$
    end if
    Normalization: Re-weight particles to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$
    Estimate target position $\tilde{\mathbf{x}}_k = \sum_{n=1}^{N} w_k^{(n)} \mathbf{x}_k^{(n)}$
    Resampling: Generate $\mathbf{x}_k^{(n)}$ from the set $\{\mathbf{x}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}$
    $k = k + 1$
end while

ing is ignored. That classification information is provided by the audio source localization algorithm [21]. For this and the remaining four methods, the classification is used.

The modified algorithm AV-PF-2D-SNS is presented as pseudo code in Table 4.4.

4.6 Particle Filter Based Audio-Visual Fusion Technique in 3-D(AV-PF-1CAM-3D)

In this method, the 2-D tracking technique AV-PF-2D-SNS described in previous section is extended to 3-D space. For this extension, the state vector is redefined with new state elements as:

$$x = [\ x_1\ \dot{x}_1\ x_2\ \dot{x}_2\ x_3\ \dot{x}_3\ s\ ]^T \tag{4.21}$$

In this equation, the positions on x, y, and z axes are denoted by x_1, x_2, and x_3 respectively. Moreover, the velocities on these axes are $\dot{x}_1$, $\dot{x}_2$, and $\dot{x}_3$ respectively. In addition to these, s is the scale of the rectangle centered around the projected particle on the image plane. Matrices of the motion model in Equation 4.12 are modified and named as $F_{3D-1cam}$, G_{3D_1cam}, and $Q_{3D-1cam}$. The matrices in 4.13 and 4.14 becomes:

$$\mathbf{F}_{3D-1cam} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{G}_{3D-1cam} = \begin{bmatrix} \frac{T^2}{2} \\ T \\ \frac{T^2}{2} \\ T \\ \frac{T^2}{2} \\ T \\ 1 \end{bmatrix} \tag{4.22}$$

$$\mathbf{Q}_{3D-1cam} = \begin{bmatrix} \sigma^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_s^2 \end{bmatrix} \tag{4.23}$$

Thus, Equation 4.12 becomes:

$$\mathbf{x}_k^{(n)} = \mathbf{F}_{\mathbf{3D-1cam}}\mathbf{x}_{k-1}^{(n)} + \mathbf{q}_{3D-1cam,k}^{(n)} \tag{4.24}$$

In this method, since the state vectors in 3-D are used, the weighting procedure of the audio components is changed. Firstly, the approach in Figure 4.2 is applied to determined coordinates on the x-y plane. After that, for each particle, the distance to the estimated location in the x-y plane is calculated. Consequently, Equation 4.17 becomes:

$$D_{audio}^{(n)} = \sqrt{(x_{particle,pos}^{(n)} - x_{audio,est})^2 + (y_{particle,pos}^{(n)} - y_{audio,est})^2} \tag{4.25}$$

In this equation, $x_{particle,pos}^{(n)}$ and $y_{particle,pos}^{(n)}$ are n-th particle position in x and y coordinates in 3-D external world. Also, $x_{audio,est}$ and $y_{audio,est}$ are estimated locations of the particle on x and y axes on the coordinate system.

AV-PF-1CAM-3D algorithm explained in this section is shown as pseudo code in Table 4.7.

4.7 Particle Filter Based Visual Tracking Technique in 3-D by Using Two Cameras(V-PF-2CAM)

In this method, the audio tracker part is excluded and the tracker is implemented using only two cameras. The aim of this implementation is to compare the visual-

Table4.5: Algorithm of the particle filter based audio-visual fusion in 3-D(AV-PF-1CAM-3D).

Initialize $N, \sigma^2, U, T, \mathbf{F}, \lambda, \lambda_{audio}, \tau, r(u), \mathbf{x}_0^{(n)}, w_0^{(n)}, k$
while $k < K$ **do**
  Propagate particles: $\mathbf{x}_k^{(n)} = \mathbf{F}_{3D-1cam}\mathbf{x}_{k-1}^{(n)} + \mathbf{G}_{3D-1cam,k}\mathbf{q}_{3D-1cam,k}^{(n)}$
  Calculate $D^{(n)}$ for visual weights using Equation 4.3
  Calculate visual weights: $w_{visual,k}^{(n)} = e^{-\lambda(D^{(n)})^2}$, for $n = 1, ..., N$
  For audio component, use crossing of the azimuth angles from two microphone arrays as it is described in Figure 4.2 in 3-D.
  Calculate $D_{audio}^{(n)}$ for audio using Equation 4.25
  Calculate audio weights: $w_{audio,k}^{(n)} = e^{-\lambda audio(D_{audio}^{(n)})^2}$, for $n = 1, ..., N$
  Average of the audio measurements: $w_{av,audio} = \frac{\sum_{i=1}^{N} w_{audio,k}^{(n)}}{N}$
  if The speaker is not silent **then**
    if $w_{av,audio} > \tau$ **then**
      $w_k^{(n)} = w_{audio,k}^{(n)} * w_{visual,k}^{(n)}$
    else
      $w_k^{(n)} = w_{visual,k}^{(n)}$
    end if
  else
    $w_k^{(n)} = w_{visual,k}^{(n)}$
  end if
  Normalization: Re-weight particles to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$
  Estimate target position $\tilde{\mathbf{x}}_k = \sum_{n=1}^{N} w_k^{(n)}\mathbf{x}_k^{(n)}$
  Resampling: Generate $\mathbf{x}_k^{(n)}$ from the set $\{\mathbf{x}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}$
  $k = k + 1$
end while

only methods and audio-visual methods.

In order to implement a tracking algorithm with two cameras, the content of the particle state in 4.21 is redefined as by adding a new scale factor s_2 for the second camera. Thus, the new state vector becomes:

$$x = [\ x_1\ \dot{x}_1\ x_2\ \dot{x}_2\ x_3\ \dot{x}_3\ s_1\ s_2\ ]^T$$

Redefinition of the state vector causes to modify matrices used in the state transition equation. Hence, the matrices in Equation 4.22 and 4.23 are redefined as:

$$\mathbf{F}_{3D-2cam} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{G}_{3D-2cam} = \begin{bmatrix} \frac{T^2}{2} \\ T \\ \frac{T^2}{2} \\ T \\ \frac{T^2}{2} \\ T \\ 1 \\ 1 \end{bmatrix} \tag{4.26}$$

$$\mathbf{Q}_{3D-2cam} = \begin{bmatrix} \sigma^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{s,1}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{s,2}^2 \end{bmatrix} \tag{4.27}$$

Hence, the motion model becomes:

$$\mathbf{x}_k^{(n)} = \mathbf{F}_{3D-2cam}\mathbf{x}_{k-1}^{(n)} + \mathbf{G}_{3D-2cam,k}\mathbf{q}_{3D-2cam,k}^{(n)} \tag{4.28}$$

Table4.6: Algorithm of the particle filter-based visual tracking technique in 3-D using two cameras(V-PF-2CAM).

1:	Initialize $N, \sigma^2, U, T, \mathbf{F}, \lambda, r(u), \mathbf{x}_0^{(n)}, w_0^{(n)}, k$
2:	**while** $k < K$ **do**
3:	Propagate particles: $\mathbf{x}_k^{(n)} = \mathbf{F}_{3D-2cam}\mathbf{x}_{k-1}^{(n)} + \mathbf{G}_{3D-2cam,k}q_{3D-2cam,k}^{(n)}$
4:	Calculate $D_1^{(n)}$ and $D_2^{(n)}$ for visual tracker using Equation 4.3
5:	Calculate the first visual weight as: $w_{visual,k,1}^{(n)} = e^{-\lambda(D_1^{(n)})^2}$, for $n = 1, ..., N$
6:	Calculate the second visual weight as: $w_{visual,k,2}^{(n)} = e^{-\lambda(D_2^{(n)})^2}$, for $n = 1, ..., N$
7:	Calculate the final weight as: $w_k^{(n)} = w_{visual,k,1}^{(n)} * w_{visual,k,2}^{(n)}$
8:	Normalization: Re-weight particles to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$
9:	Estimate target position $\tilde{x}_k = \sum_{n=1}^{N} w_k^{(n)}\mathbf{x}_k^{(n)}$
10:	Resampling: Generate $\mathbf{x}_k^{(n)}$ from the set $\{\mathbf{x}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}$
11:	$k = k + 1$
12:	**end if**

Equation 4.16 is modified for two camera case and the new formula about weighting becomes:

$$w_k^{(n)} = e^{-\lambda(D_{cam1}^{(n)})^2} * e^{-\lambda(D_{cam2}^{(n)})^2} \tag{4.29}$$

The V-PF-2CAM algorithm explained in this section is given as pseudo code in Table 4.6.

4.8 Particle Filter Based Audio-Visual Fusion Technique in 3-D by Using Two Cameras and Two Microphone Arrays(AV-PF-3D)

For AV-PF-3D, the audio measurements from two microphone arrays are integrated to V-PF-2CAM method that is explained in Section 4.7.

The new algorithm with two cameras and two microphone arrays is given as pseudo code in Table 4.7.

Table4.7: Algorithm of the particle filter based audio-visual fusion in 3-D using two cameras and two microphone arrays(AV-PF-3D).

Initialize $N, \sigma^2, U, T, \mathbf{F}, \lambda, \lambda_{audio}, \tau, r(u), \mathbf{x}_0^{(n)}, w_0^{(n)}, k$
while $k < K$ **do**
  Propagate particles: $\mathbf{x}_k^{(n)} = \mathbf{F}_{3D-1cam}\mathbf{x}_{k-1}^{(n)} + \mathbf{G}_{3D-1cam,k}\mathbf{q}_{3D-1cam,k}^{(n)}$
  Calculate $D_1^{(n)}$ and $D_2^{(n)}$ for visual tracker using Equation 4.3
  Calculate the first visual weight as: $w_{visual,k,1}^{(n)} = e^{-\lambda(D_1^{(n)})^2}$, for $n = 1, ..., N$
  Calculate the second visual weight as: $w_{visual,k,2}^{(n)} = e^{-\lambda(D_2^{(n)})^2}$, for $n = 1, ..., N$
  Calculate the weight of the visual part as: $w_{visual,k}^{(n)} = w_{visual,k,1}^{(n)} * w_{visual,k,2}^{(n)}$
  For audio component, use crossing of the azimuth angles from two microphone arrays as it is described in Figure 4.2 in 3-D.
  Calculate $D_{audio}^{(n)}$ for audio using Equation 4.25
  Calculate audio weights: $w_{audio,k}^{(n)} = e^{-\lambda_{audio}(D_{audio}^{(n)})^2}$, for $n = 1, ..., N$
  Average of the audio measurements: $w_{av,audio} = \frac{\sum_{i=1}^{N} w_{audio,k}^{(n)}}{N}$
  if The speaker is not silent **then**
    if $w_{av,audio} > \tau$ **then**
      $w_k^{(n)} = w_{audio,k}^{(n)} * w_{visual,k}^{(n)}$
    else
      $w_k^{(n)} = w_{visual,k}^{(n)}$
    end if
  else
    $w_k^{(n)} = w_{visual,k}^{(n)}$
  end if
  Normalization: Re-weight particles to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$
  Estimate target position $\tilde{\mathbf{x}}_k = \sum_{n=1}^{N} w_k^{(n)}\mathbf{x}_k^{(n)}$
  Resampling: Generate $\mathbf{x}_k^{(n)}$ from the set $\{\mathbf{x}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}$
  $k = k + 1$
end while

4.9 Particle Filter Based Audio-Visual Fusion Tracking Technique in 3-D by Using Two Cameras and Two Microphone Arrays with Occlusion Handling(AV-PF-RAND)

In this tracking method, AV-PF-3D technique presented in Section 4.8 is modified so that it can handle the occlusion situations more efficiently. Also, it is predicted that these modifications also increase the robustness of the tracking method in terms of accuracy. These modifications in the method contain two major part:

1. Random particle injection
2. Discarding of the unreliable visual data

In order to prevent particle deprivation problem, the random particle injection technique is implemented. Details of the particle deprivation problem and Section 2.5.4.5. Also, the implementation details about random particle injection technique is presented in Table 2.4 in this section.

Additionally, in order to increase the accuracy of the tracking system, unreliable visual data is discarded from both cameras. As it is stated previously, the unreliable audio data or the audio data having low weight is discarded using τ parameter. For the visual data, κ parameter is used as threshold value. In the case that all the sensor data is discarded, the states of the particles are updated only using the motion model.

The explained AV-PF-RAND algorithm is shown as pseudo code in Table 4.8.

1:	Initialize $N, \sigma^2, U, T, \mathbf{F}, \lambda, \lambda_{audio}, \tau, \kappa, \alpha_{slow}, \alpha_{fast}, r(u), \mathbf{x}_0^{(n)}, w_0^{(n)}, k$
2:	**while** $k < K$ **do**
3:	Propagate particles: $\mathbf{x}_k^{(n)} = \mathbf{F}_{3D-1cam}\mathbf{x}_{k-1}^{(n)} + \mathbf{G}_{3D-1cam,k}\mathbf{q}_{3D-1cam,k}^{(n)}$
4:	Calculate $D_1^{(n)}$ and $D_2^{(n)}$ for visual tracker using Equation 4.3
5:	Calculate the first visual weight as: $w_{visual,k,1}^{(n)} = e^{-\lambda(D_1^{(n)})^2}$, for $n = 1, \ldots, N$
6:	Average of the video measurements: $w_{av,visual,1} = \frac{\sum_{i=1}^{N} w_{visual,k,1}^{(n)}}{N}$
7:	**if** $w_{av,visual,1} > \kappa$ **then**
8:	$w_{visual,k}^{(n)} = w_{visual,k,1}^{(n)}$ for $n = 1, \ldots, N$

else

$w_{visual,k}^{(n)} = w_0^{(n)}$

end if

Calculate the second visual weight as: $w_{visual,k,2}^{(n)} = e^{-\lambda(D_2^{(n)})^2}$, for $n = 1, \ldots, N$

Average of the video measurements: $w_{av,visual,1} = \frac{\sum_{i=1}^{N} w_{visual,k,2}^{(n)}}{N}$

if $w_{av,visual,2} > \kappa$ **then**

$w_{visual,k}^{(n)} = w_{visual,k}^{(n)} * w_{visual,k,2}^{(n)}$ for $n = 1, \ldots, N$

else

$w_{visual,k}^{(n)} = w_{visual,k}^{(n)}$ for $n = 1, \ldots, N$

end if

For audio component, use crossing of the azimuth angles from two microphone arrays as it is described in Figure 4.2 in 3-D.

Calculate $D_{audio}^{(n)}$ for audio using Equation 4.25

Calculate audio weights: $w_{audio,k}^{(n)} = e^{-\lambda_{audio}(D_{audio}^{(n)})^2}$, for $n = 1, \ldots, N$

Average of the audio measurements: $w_{av,audio} = \frac{\sum_{i=1}^{N} w_{audio,k}^{(n)}}{N}$

if The speaker is not silent **then**

if $w_{av,audio} > \tau$ **then**

$w_k^{(n)} = w_{audio,k}^{(n)} * w_{visual,k}^{(n)}$

else

$w_k^{(n)} = w_{visual,k}^{(n)}$

end if

else

$w_k^{(n)} = w_{visual,k}^{(n)}$

end if

$w_{avg} = w_{avg} + \frac{1}{N} w_t^{[n]}$

$w_{slow} = w_{slow} + \alpha_{slow}(w_{avg} - w_{slow})$

$w_{fast} = w_{fast} + \alpha_{fast}(w_{avg} - w_{fast})$

for $n = 1\ to\ N$ **do**

with $probability\ max\{0.0, 1.0 - w_{fast}/w_{slow}\}$ **do**

37: $add\ random\ pose\ to\ \chi_t$

38: **else**

39: $draw\ i \in \{1, \ldots, N\}\ with\ probability \propto w_t^{[i]}$

40: $add\ mathbf{x}_t^{[i]}\ to\ \chi_t$

41: **end with**

42: **end for**

43: Normalization: Re-weight particles to ensure that $\sum_{n=1}^{N} w_k^{(n)} = 1$

44: Estimate target position $\tilde{\mathbf{x}}_k = \sum_{n=1}^{N} w_k^{(n)} \mathbf{x}_k^{(n)}$

45: Resampling: Generate $\mathbf{x}_k^{(n)}$ from the set $\{\mathbf{x}_k^{(n)}, w_k^{(n)}\}_{n=1}^{N}$

46: $k = k + 1$

47: **end**

Table4.8: Algorithm of the particle filter based audio-visual fusion in 3-D using two cameras and two microphone arrays with occlusion handling(AV-PF-RAND).

In this chapter, all the implemented methods are explained in a detailed way. In the following chapter, implementation results of these eight algorithms are presented.

CHAPTER 5

RESULTS OF THE IMPLEMENTATIONS OF TRACKING METHODS

5.1 Introduction

Eight particle filter based person tracking algorithms are explained in Chapter 4. In this Chapter, the simulation results of the implementations are presented. Firstly, the simulation details and choices for parameters are explained. Following to that, the tracking results of 2-D and 3-D tracking cases in terms of pixel or distance error and success rate of tracking are demonstrated in the related tables and reasons for these results are analyzed. Also, the graphical results of these tracking are presented in Appendix. Lastly, 2-D and 3-D methods are compared.

5.2 Implementation Details and Parameter Settings

During the simulations, MATLAB is used as the implementation tool and all numerical and graphical results are prepared with it. For audio source localization, open-source MATLAB codes by Lathoud [21] is used. Audio localization results of the algorithm [21] are used as input in our implementation. For visual parts and fusion parts in the particle filtering framework, we have prepared the required codes.

First four methods are implementation in 2-D face tracking system, while the remaining four are working in 3-D. Hence, results are presented in two divided categories as 2-D tracker and 3-D tracker. For 2-D implementations, each camera is used sepa-

rately, while for 3-D implementations, two of three cameras are used.

Graphical results for both type of trackers are presented in Appendix as it is stated previously. In these graphs, the tracking error is illustrated for the related sequence for each frame. Here, the error at frame k is given as the average of the errors from frame 1 to k. This representation is preferred instead of plotting error on corresponding frame k, which would give oscillating graph since errors may change abruptly in subsequent frames. In addition to this, plotting average error at each frame k gives smooth graph which can be interpreted easily and the overall performance of each tracker can be compared clearly [18]. In Appendix, the results for AV-PF-1CAM-3D is not presented. It fails to track and hence error values are too big. Thus, the results for this method are excluded from the graphical results to get a better view.

Numerical results for both type trackers contains two metrics. First metric is MAE(mean absolute error). In this metric, for 2-D cases, the sum of Euclidean distance in pixels between the estimated and the ground truth positions are calculated. But, for 3-D cases, Euclidean distances in terms of meters are summed. Then, the sum is divided by number of frames [18]. Second metric is the success rate of tracking. For 2-D cases, if the distance of the estimated target is more than 30 pixel for 5 seconds, then it is assumed the target is lost. For 3-D cases, the distance limit is 0.4 m.

Parameters in all of the methods are chosen to observe optimum results. After applying enough number of tries, parameters are selected. Chosen parameters for each methods are given in Table 5.1. All of the parameters listed in this table are explained in the related section of Chapter 4.

Furthermore, all simulations are repeated 10 times for each experiment.

Following sections presents the implementation results and analyses of them. Total seven different sequences are used in implementations. Description of the these sequences are presented in Section 3.5. There are single, two and three person cases in these sequences. For all these cases, MAE and success rate of tracking are presented side by side for each camera and method. The best results at each camera and at overall results of tracking methods are indicated with yellow background for all the tables. Additionally, "NA" terms in these tables stands for "Not Applicable"

	N: Number of Particles	σ^2: Variance of position and velocity	σ_s^2: Variance of scale factor	U: Number of histogram bins	T: Period the image frame (in seconds)	λ: Parameter to adjust weighting of visual cues	λ_{audio}: Parameter to set weighting of audio cues
V-PF	125	50	$4x10^{-4}$	8	0.04	150	
AV-PF	125	50	$4x10^{-4}$	8	0.04	150	
AV-PF-2D	125		$4x10^{-4}$	8	0.04	150	$5x10^{-2}$
AV-PF-2D-1CAM	125		$4x10^{-4}$	8	0.04	150	$5x10^{-2}$
AV-PF-1CAM-3D	125		$4x10^{-4}$	8	0.04	150	150
V-PF-2CAM	125		$4x10^{-4}$	8	0.04	150	150
AV-PF-3D	125		$4x10^{-4}$	8	0.04	150	150
AV-PF-RAND	125		$4x10^{-4}$	8	0.04	150	150

	σ^2: Variance of position and velocity for the new proposed motion model	τ: Parameter to discard unrelated audio data	κ: Parameter to discard unrelated visual data	α_{slow}: Parameter to estimate long term average for random particle injection	α_{fast}: Parameter to estimate long term average for random particle injection
V-PF					
AV-PF					
AV-PF-2D	$5x10^{-6}$	10^{-2}			
AV-PF-2D-1CAM	$5x10^{-6}$	10^{-2}			
AV-PF-1CAM-3D	10^{3}	10^{-4}			
V-PF-2CAM	10^{3}				
AV-PF-3D	10^{3}	10^{-4}			
AV-PF-RAND	10^{3}	10^{-4}	$3.8x10^{-11}$	10^{-8}	10^{-6}

Table5.1: Values of Parameters in the Implementation of Tracking Methods.

5.3 Results of 2-D Tracking Algorithms

5.3.1 Single Person Case

In Table 5.2, MAE results and the success rate of tracking for single person case are presented.

	seq01-1p-0000							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF-2D	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF-2D-SNS	NA	0%	NA	0%	NA	0%	NA	0%
	seq11-1p-0100							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	14.98	100%	15.07	100%	18.68	100%	16.24	100%
AV-PF	15.22	100%	32.79	80%	18.39	100%	22.13	93%
AV-PF-2D	14.92	100%	15.37	100%	19.26	100%	16.52	100%
AV-PF-2D-SNS	15.27	100%	15.36	100%	19.31	100%	16.65	100%
	seq15-1p-0100							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	13.25	100%	15.99	90%	33.98	40%	21.07	76.67
AV-PF	13.51	100%	12.05	100%	30.46	60%	18.67	86.67
AV-PF-2D	13.85	100%	13.01	100%	31.19	20%	19.35	73.33
AV-PF-2D-SNS	14.2	100%	12.59	100%	39.71	10%	22.17	70

Table5.2: Results of 2-D Tracking Methods For Single Person Case. MAE is shown in terms of pixels and the success rate of tracking is shown as percentage.

For Sequence #1, all the 2-D trackers are failed. Although the speaker is stable while speaking, he rotates during the sequence. In the implemented methods, a single predetermined image of the target head is chosen for tracking. Hence, if the speaker turns the back of her/his head to the camera in way that is not similar to the predetermined image, then particles deprive as it is observed in the results of sequence #1. Also, if the target stands in a environment in which the head of the speaker can not be distinguishable from the background, then the tracker fails. An example for this

situation can be seen in Figure 5.1.

Figure 5.1: An example for the situation that the speaker's head can not be distinguishable from the background in seq01-1p-0000 for camera #2.

For Sequence #11, all the methods are successful at tracking, although the speaker makes abrupt movements in this sequence. In overall, V-PF performs better. But, AV-PF-2D and AV-PF-2D-SNS performs nearly same to it.

For Sequence #15, the tracking performance of AV-PF is the best. Long silence periods of this sequence make performances of AV-PF-2D and AV-PF-2D-SNS worse.

5.3.2 Two Person Case

In Table 5.3, MAE results and the success rate of tracking for single person case are presented.

For Sequence #18 and #24, V-PF fails. Since, the tracking using only one visual cue is not a proper approach for multi-person tracking. However, AV-PF and AV-PF-2D performs well for these two person cases.

These methods are not efficient, if speakers are out of the field of view. This effect can be seen in Sequence #24 for both person. In this sequence, both speakers are out of field of the camera view and the trackers fail. These situation can be seen on Figure A.6 and A.7 for both speakers at camera #1.

	seq18-2p-0101-person1							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF	15.72	90%	17.03	100%	14.93	100%	15.89	97%
AV-PF-2D	19.05	40%	17.23	100%	22.5	90%	19.59	77%
AV-PF-2D-SNS	20.74	10%	16.83	100%	15.58	90%	17.72	67%
	seq18-2p-0101-person2							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF	12.32	100%	15.93	100%	22.18	100%	16.81	100%
AV-PF-2D	20.35	100%	14.35	100%	21.15	100%	18.62	100%
AV-PF-2D-SNS	23.78	80%	14.55	100%	21.07	100%	19.8	93%
	seq24-2p-0111-person1							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF	NA	0%	29.27	10%	33	10%	31.14	7%
AV-PF-2D	NA	0%	31.86	10%	NA	0%	31.86	3%
AV-PF-2D-SNS	NA	0%	32.24	10%	NA	0%	32.24	3%
	seq24-2p-0111-person2							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	NA	0%	NA	0%	NA	0%	NA	0
AV-PF	NA	0%	34.81	10%	NA	0%	34.81	10%
AV-PF-2D	NA	0%	35.39	20%	NA	0%	35.39	20%
AV-PF-2D-SNS	NA	0%	35.75	20%	NA	0%	35.75	20%

Table5.3: Results of 2-D Tracking Methods For Two Person Case. MAE is shown in terms of pixels and the success rate of tracking is shown as percentage.

5.3.3 Three Person Case

In Table 5.4, MAE results and the success rate of tracking for single person case are presented.

	seq40-3p-0111-person1							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	5.02	100%	10.78	100%	22.57	50%	12.79	83%
AV-PF	12.79	80%	9.9	100%	18.27	60%	13.65	80%
AV-PF-2D	5.04	100%	20.69	80%	19.13	70%	14.95	83%
AV-PF-2D-SNS	5.03	100%	25.77	70%	17.63	90%	16.14	87%
	seq40-3p-0111-person2							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF	15.65	100%	20.05	70%	30.81	10%	22.17	60%
AV-PF-2D	32	80%	11.03	100%	NA	0%	21.52	60%
AV-PF-2D-SNS	13.84	100%	13.35	90%	NA	0%	13.6	63%
	seq40-3p-0111-person3							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	10.01	100%	5.55	100%	8.61	100%	8.06	100%
AV-PF	11.79	100%	6.51	100%	8.72	100%	9.01	100%
AV-PF-2D	10.33	100%	5.86	100%	9.18	100%	8.46	100%
AV-PF-2D-SNS	10.42	100%	5.75	100%	9.59	100%	8.59	100%
	seq45-3p-1111-person1							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF	14.84	100%	10.19	100%	20.25	50%	15.09	83%
AV-PF-2D	15.84	100%	15.81	90%	34.58	10%	22.08	67%
AV-PF-2D-SNS	15.26	100%	20.74	100%	NA	0%	12	67%
	seq45-3p-1111-person2							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF	47.4	90%	NA	0%	43.96	90%	45.68	60%
AV-PF-2D	46.72	90%	NA	0%	NA	0%	46.72	30%
AV-PF-2D-SNS	45.44	90%	NA	0%	NA	0%	45.44	30%
	seq45-3p-1111-person3							
	Camera #1		Camera #2		Camera #3		Overall	
V-PF	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF	11.69	100%	30.4	80%	NA	0%	21.05	60%
AV-PF-2D	12.31	100%	15.17	100%	42.17	80%	34.83	93%
AV-PF-2D-SNS	11.64	100%	15	100%	28.91	100%	18.52	100%

Table5.4: Results of 2-D Tracking Methods For Three Person Case. MAE is shown in terms of pixels and the success rate of tracking is shown as percentage.

In Sequence #40, two person are sitting and one person, namely person #2, walks around them. Also, all of them speak in the sequence. Additionally, two standing person does not make any movement and do not turn their heads. For stable targets,

V-PF performs well. However, for walking person #2, V-PF fails as it is observed in the similar scenarios of moving multiple targets. For moving person, AV-PF-2D-SNS performs the best with speech/non-speech classification.

5.3.4 Conclusion

Although V-PF method performs well for single target sequences, it fails to track moving targets for multi-target cases. Since, for these cases, the usage of only one visual cue results that particles are distributed around the faces of the other trackers also. If the speaker rotates her/his head, weights of these particles around the target decreases. However, the weights of particles around the other faces can be stood relatively high with respect to these particles. Hence, in the resampling stage, particles with lower values may disappear. Consequently, the particles will be stuck around wrong faces and this will cause to losing of the target.

AV-PF performs well except Sequence #1. However, AV-PF-2D and AV-PF-2D-SNS shows better performances especially for multi-target cases. In AV-PF, two separate estimates are generated by using only visual cues and audio-visual cues. After this generation, the tracking estimate with higher Bhattacharyya distance is used. Depending the predetermined images of the speakers and the condition of the speaker's head rotation, particles may move around the wrong faces. As it is described in the previous paragraph, for a specific moment, this may cause to losing of the target. Additionally, it decreases the accuracy of the tracker. However, in AV-PF-2D and AV-PF-2D-SNS, particles are moved using both audio and visual cues together. The complementary nature of these two data results that in the case of one these measurement is not reliable enough, the other cue affects the system so that particles can track the target.

5.4 Results of 3-D Tracking Algorithms

AV-PF-1CAM-3D method fails for all sequences. In other words, tracking rate is 0% for all sequences. Hence, the tracking results of that method is not presented neither in this section nor in Appendix part. Usage of only one camera is not enough to

propagate particles to the true locations of targets.

5.4.1 Single Person Case

In Table 5.5, MAE results and the success rate of tracking for single person case are presented.

	seq01-1p-0000							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF-3D	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF-RAND	NA	0%	NA	0%	NA	0%	NA	0%
	seq11-1p-0100							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	0.14	100%	0.2	100%	0.17	100%	0.17	100%
AV-PF-3D	0.13	100%	0.2	100%	0.17	100%	0.17	100%
AV-PF-RAND	0.13	100%	0.22	100%	0.17	100%	0.17	100%
	seq15-1p-0100							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	0.17	100%	0.4	100%	0.19	100%	0.25	100%
AV-PF-3D	0.16	100%	0.35	100%	0.19	100%	0.23	100%
AV-PF-RAND	0.17	100%	0.35	100%	0.3	90%	0.27	97%

Table5.5: Results of 3-D Tracking Methods For Single Person Case. MAE is shown in terms of meters and the success rate of tracking is shown as percentage.

3-D trackers fail to follow targets in Sequence #1 as 2-D trackers. Same reasons for 2-D trackers are also valid for 3-D cases. Additionally, if the visual cue on the one of the camera is distorted, then the overall tracker fails.

For Sequence #11 and #15, although AV-PF-3D shows the best performance. However, other trackers show similar performances also.

5.4.2 Two Person Case

In Table 5.6, MAE results and the success rate of tracking for single person case are presented.

	seq18-2p-0101-person1							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	0.19	10%	0.19	100%	0.22	10%	0.2	40%
AV-PF-3D	0.17	20%	0.19	100%	0.18	30%	0.18	50%
AV-PF-RAND	0.17	20%	0.19	100%	0.17	30%	0.18	50%
	seq18-2p-0101-person2							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	0.24	100%	0.17	100%	0.2	100%	0.2	100%
AV-PF-3D	0.26	100%	0.22	100%	0.22	100%	0.23	100%
AV-PF-RAND	0.26	100%	0.22	100%	0.23	100%	0.24	100%
	seq24-2p-0111-person1							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF-3D	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF-RAND	NA	0%	NA	0%	NA	0%	NA	0%
	seq24-2p-0111-person2							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF-3D	NA	0%	NA	0%	NA	0%	NA	0%
AV-PF-RAND	NA	0%	NA	0%	NA	0%	NA	0%

Table5.6: Results of 3-D Tracking Methods For Two Person Case. MAE is shown in terms of meters and the success rate of tracking is shown as percentage.

For Sequence #18, all methods track their targets. Also, the performance of all methods are similar. However, 3-D trackers fail at Sequence #24. Because, in this sequence, speakers moves out the field of views of the cameras and distorted measurement values in one of the camera results to losing of the target.

5.4.3 Three Person Case

In Table 5.7, MAE results and the success rate of tracking for single person case are presented.

As expected, 3-D trackers perform better than 2-D trackers. Although, in some cases, AV-PF-RAND loses the target, V-PF-2CAM and AV-PF-RAND shows similar performances for the sequences.

5.4.4 Conclusion

In some cases, AV-PF-RAND improves the performance. Since, it recovers particles from wrong states by adding random particles. However, for some cases, that may also results that particles are moved around the wrong states.

By comparing the performance of V-PF-2CAM and AV-PF-3D, it can be concluded that if there are no strict visual occlusions in the scene, there is no significant gains using audio-visual fusion instead of two cameras only. Obviously, for our case, it is related with the noisy audio estimates for the target locations. By improving audio tracker or estimating the target location in 3-D space, audio-visual fusion probably will give better results.

5.5 Comparison of 2-D Trackers and 3-D Trackers

Errors of 3-D trackers are projected to 2-D image planes so that that results of 2-D and 3-D trackers becomes comparable.

	seq40-3p-0111-person1							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	0.07	100%	0.36	30%	0.2	100%	0.21	77%
AV-PF-3D	0.07	100%	0.4	20%	0.19	100%	0.22	73%
AV-PF-RAND	NA	0%	NA	0%	NA	0%	NA	0%
	seq40-3p-0111-person2							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	0.22	100%	0.27	80%	0.13	100%	0.21	93%
AV-PF-3D	0.11	100%	0.32	20%	0.13	100%	0.19	73%
AV-PF-RAND	0.11	100%	0.28	40%	0.15	90%	0.18	77%
	seq40-3p-0111-person3							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	0.08	100%	0.09	100%	0.08	100%	0.08	100%
AV-PF-3D	0.08	100%	0.09	100%	0.08	100%	0.08	100%
AV-PF-RAND	0.12	80%	0.09	100%	0.08	80%	0.1	87%
	seq45-3p-1111-person1							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	0.39	60%	0.27	100%	0.25	100%	0.3	87%
AV-PF-3D	0.42	40%	0.26	100%	0.28	70%	0.32	70%
AV-PF-RAND	NA	0%	NA	0%	NA	0%	NA	0%
	seq45-3p-1111-person2							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	1.34	10%	NA	0%	0.96	10%	1.15	7%
AV-PF-3D	NA	0%	NA	0%	1.04	10%	1.04	3%
AV-PF-RAND	1.09	10%	NA	0%	NA	0%	1.09	3%
	seq45-3p-1111-person3							
	Camera #1 & #2		Camera #2 & #3		Camera #1 & #3		Overall	
V-PF-2CAM	0.2	70%	0.24	100%	0.33	70%	0.26	80%
AV-PF-3D	0.24	100%	0.27	90%	0.38	70%	0.3	87%
AV-PF-RAND	0.16	100%	0.25	100%	0.38	80%	0.26	93%

Table5.7: Results of 3-D Tracking Methods For Three Person Case. MAE is shown in terms of meters and the success rate of tracking is shown as percentage.

5.5.1 Single Person Case

In Table 5.8, MAE results in terms of pixels and success rate of tracking for single person case are presented. Results for Sequence #1 are not presented. Since, all tracker fails for this sequence.

	Used Cameras for 3-D Trackers	seq11-1p-0100							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		14.98	100%	15.07	100%	18.68	100%	16.24	100%
AV-PF		15.22	100%	32.79	80%	18.39	100%	22.13	93%
AV-PF-2D		14.92	100%	15.37	100%	19.26	100%	16.52	100%
AV-PF-2D-SNS		15.27	100%	15.36	100%	19.31	100%	16.65	100%
V-PF-2CAM	1&2	15.57	100%	15	100%			18.16	100%
V-PF-2CAM	2&3			17.21	100%	21.79	100%		
V-PF-2CAM	1&3	17.4	100%			21.96	100%		
AV-PF-3D	1&2	15.49	100%	14.84	100%			18.16	100%
AV-PF-3D	2&3			17.18	100%	21.68	100%		
AV-PF-3D	1&3	17.54	100%			22.23	100%		
AV-PF-RAND	1&2	15.56	100%	14.9	100%			18.39	100%
AV-PF-RAND	2&3			18.23	100%	22.11	100%		
AV-PF-RAND	1&3	17.35	100%			22.19	100%		
		seq15-1p-0100							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		13.25	100%	15.99	90%	33.98	40%	21.07	77%
AV-PF		13.51	100%	12.05	100%	30.46	60%	18.67	87%
AV-PF-2D		13.85	100%	13.01	100%	31.19	20%	19.35	73%
AV-PF-2D-SNS		14.2	100%	12.59	100%	39.71	10%	22.17	70%
V-PF-2CAM	1&2	13.55	100%	17.21	100%			15.04	100%
V-PF-2CAM	2&3			12.17	100%	13.04	100%		
V-PF-2CAM	1&3	14.46	100%			19.8	100%		
AV-PF-3D	1&2	17.67	100%	13.29	100%			14.95	100%
AV-PF-3D	2&3			11.91	100%	12.32	100%		
AV-PF-3D	1&3	14.61	100%			19.91	100%		
AV-PF-RAND	1&2	13.68	100%	18.18	100%			18.35	95%
AV-PF-RAND	2&3			11.81	100%	11.96	100%		
AV-PF-RAND	1&3	15.42	90%			39.06	80%		

Table5.8: Overall Results for Single Person Tracking. MAE is shown in terms of meters and the success rate of tracking is shown as percentage.

For Sequence #11, all sequences performs similar. However, with the usage of the second camera, 3-D trackers perform better for Sequence #15 with long silence duration.

5.5.2 Two Person Case

In Table 5.9, MAE results in terms of pixels and success rate of tracking for two person case are presented. Results for Sequence #24 are not presented. Since, all 3-D tracker fails for this sequence.

	Used Cameras for 3-D Trackers	seq18-2p-0101-person1							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		NA	0%	NA	0%	NA	0%	NA	0%
AV-PF		15.72	90%	17.03	100%	14.93	100%	15.89	97%
AV-PF-2D		19.05	40%	17.23	100%	22.5	90%	19.59	77%
AV-PF-2D-SNS		20.74	10%	16.83	100%	15.58	90%	17.72	67%
V-PF-2CAM	1&2	22.46	10%	15.67	90%			17.6	68%
V-PF-2CAM	2&3			15.52	100%	16.88	100%		
V-PF-2CAM	1&3	21.74	10%			13.34	100%		
AV-PF-3D	1&2	21.93	20%	14.83	100%			17.58	73%
AV-PF-3D	2&3			15.53	100%	16.75	100%		
AV-PF-3D	1&3	20.99	30%			15.47	90%		
AV-PF-RAND	1&2	21.93	20%	14.76	100%			17.39	73%
AV-PF-RAND	2&3			15.59	100%	16.76	100%		
AV-PF-RAND	1&3	21.09	30%			14.19	90%		
		seq18-2p-0101-person2							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		NA	0%	NA	0%	NA	0%	NA	0%
AV-PF		12.32	100%	15.93	100%	22.18	100%	16.81	100%
AV-PF-2D		20.35	100%	14.35	100%	21.15	100%	18.62	100%
AV-PF-2D-SNS		23.78	80%	14.55	100%	21.07	100%	19.8	93%
V-PF-2CAM	1&2	20.87	100%	12.64	100%			17.75	100%
V-PF-2CAM	2&3			13.39	100%	21.08	100%		
V-PF-2CAM	1&3	16.79	100%			21.73	100%		
AV-PF-3D	1&2	22.93	100%	12.54	100%			18.46	100%
AV-PF-3D	2&3			13.62	100%	21.11	100%		
AV-PF-3D	1&3	18.61	100%			21.97	100%		
AV-PF-RAND	1&2	22.88	100%	12.49	100%			18.53	100%
AV-PF-RAND	2&3			13.51	100%	21.12	100%		
AV-PF-RAND	1&3	19.56	100%			21.62	100%		

Table5.9: Overall Results for Two Person Tracking. MAE is shown in terms of meters and the success rate of tracking is shown as percentage.

For Sequence #18, AV-PF-2D shows better results. However, the performance of the remaining methods, except V-PF, are similar to AV-PF-2D.

5.5.3 Three Person Case

In Table 5.10, MAE results in terms of pixels and success rate of tracking for three person case are presented for Sequence #40.

	Used Cameras for 3-D Trackers	seq40-3p-0111-person1							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		5.02	100%	10.78	100%	22.57	50%	12.79	83%
AV-PF		12.79	80%	9.9	100%	18.27	60%	13.65	80%
AV-PF-2D		5.04	100%	20.69	80%	19.13	70%	14.95	83%
AV-PF-2D-SNS		5.03	100%	25.77	70%	17.63	90%	16.14	87%
V-PF-2CAM	1&2	6.05	100%	7.47	100%				
V-PF-2CAM	2&3			14.73	70%	15.93	80%	12.32	75%
V-PF-2CAM	1&3	5.2	100%			24.53	0%		
AV-PF-3D	1&2	6.03	100%	7.47	100%				
AV-PF-3D	2&3			15.04	70%	16.06	90%	12.11	78%
AV-PF-3D	1&3	5.32	100%			22.74	10%		
AV-PF-RAND	1&2	NA	0%	NA	0%				
AV-PF-RAND	2&3			NA	0%	NA	0%	NA	0%
AV-PF-RAND	1&3	NA	0%			NA	0%		
		seq40-3p-0111-person2							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		NA	0%	NA	0%	NA	0%	NA	0%
AV-PF		15.65	100%	20.05	70%	30.81	10%	22.17	60%
AV-PF-2D		32	80%	11.03	100%	NA	0%	21.52	60%
AV-PF-2D-SNS		13.84	100%	13.35	90%	NA	0%	13.6	63%
V-PF-2CAM	1&2	13.07	100%	9.8	100%				
V-PF-2CAM	2&3			10.87	100%	13.44	100%	12.18	100%
V-PF-2CAM	1&3	12.65	100%			13.23	100%		
AV-PF-3D	1&2	13.06	100%	9.8	100%				
AV-PF-3D	2&3			10.73	100%	13.89	100%	12.23	100%
AV-PF-3D	1&3	12.84	100%			13.05	100%		
AV-PF-RAND	1&2	13.08	100%	9.81	100%				
AV-PF-RAND	2&3			10.45	100%	13.09	100%	12.98	98%
AV-PF-RAND	1&3	17	100%			14.46	90%		
		seq40-3p-0111-person3							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		10.01	100%	5.55	100%	8.61	100%	8.06	100%
AV-PF		11.79	100%	6.51	100%	8.72	100%	9.01	100%
AV-PF-2D		10.33	100%	5.86	100%	9.18	100%	8.46	100%
AV-PF-2D-SNS		10.42	100%	5.75	100%	9.59	100%	8.59	100%
V-PF-2CAM	1&2	9.99	100%	7.84	100%				
V-PF-2CAM	2&3			8.23	100%	8.19	100%	9.02	100%
V-PF-2CAM	1&3	10.95	100%			8.94	100%		
AV-PF-3D	1&2	10.14	100%	7.89	100%				
AV-PF-3D	2&3			8.25	100%	8.21	100%	9.04	100%
AV-PF-3D	1&3	10.83	100%			8.94	100%		
AV-PF-RAND	1&2	14.91	80%	8.02	100%				
AV-PF-RAND	2&3			8.19	100%	8.14	100%	9.84	97%
AV-PF-RAND	1&3	10.85	100%			8.94	100%		

Table5.10: Overall Results for Three Person Tracking for seq40-3p-0111. MAE is shown in terms of meters and the success rate of tracking is shown as percentage.

In Table 5.11, MAE results in terms of pixels and success rate of tracking for three person case are presented for Sequence #45.

	Used Cameras for 3-D Trackers	seq45-3p-1111-person1							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		NA	0%	NA	0%	NA	0%	NA	0%
AV-PF		14.84	100%	10.19	100%	20.25	50%	15.09	83%
AV-PF-2D		15.84	100%	15.81	90%	34.58	10%	22.08	67%
AV-PF-2D-SNS		15.26	100%	20.74	100%	NA	0%	12	67%
V-PF-2CAM	1&2	10.41	100%	31.36	50%				
V-PF-2CAM	2&3			21.87	100%	19.39	100%	19.84	88%
V-PF-2CAM	1&3	16.24	100%			19.77	80%		
AV-PF-3D	1&2	11.53	90%	32.36	40%				
AV-PF-3D	2&3			20.87	100%	18.77	100%	20.7	83%
AV-PF-3D	1&3	19.86	80%			20.79	90%		
AV-PF-RAND	1&2	53.92	60%	NA	0%				
AV-PF-RAND	2&3			36.29	10%	37.13	10%	42.45	13%
AV-PF-RAND	1&3	NA	0%			NA	0%		
		seq45-3p-1111-person2							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		NA	0%	NA	0%	NA	0%	NA	0%
AV-PF		47.4	90%	NA	0%	43.96	90%	45.68	60%
AV-PF-2D		46.72	90%	NA	0%	NA	0%	46.72	30%
AV-PF-2D-SNS		45.44	90%	NA	0%	NA	0%	45.44	30%
V-PF-2CAM	1&2	113	10%	NA	0%				
V-PF-2CAM	2&3			NA	0%	NA	0%	96.49	13%
V-PF-2CAM	1&3	79.98	70%			NA	0%		
AV-PF-3D	1&2	NA	0%	NA	0%				
AV-PF-3D	2&3			NA	0%	NA	0%	91.63	10%
AV-PF-3D	1&3	91.63	60%			NA	0%		
AV-PF-RAND	1&2	86.25	60%	88.42	10%				
AV-PF-RAND	2&3			NA	0%	NA	0%	85.9	23%
AV-PF-RAND	1&3	83.04	70%			NA	0%		
		seq45-3p-1111-person3							
		Camera #1		Camera #2		Camera #3		Overall	
V-PF		NA	0%	NA	0%	NA	0%	NA	0%
AV-PF		11.69	100%	30.4	80%	NA	0%	21.05	60%
AV-PF-2D		12.31	100%	15.17	100%	42.17	80%	34.83	93%
AV-PF-2D-SNS		11.64	100%	15	100%	28.91	100%	18.52	100 %
V-PF-2CAM	1&2	12.03	100%	14.9	100%				
V-PF-2CAM	2&3			13.73	100%	20.19	100%	17.12	95%
V-PF-2CAM	1&3	12.62	100%			29.22	70%		
AV-PF-3D	1&2	17.89	100%	12.49	100%				
AV-PF-3D	2&3			13.8	100%	22.46	90%	19.03	88%
AV-PF-3D	1&3	12.15	100%			35.4	40%		
AV-PF-RAND	1&2	12.43	100%	12.3	100%				
AV-PF-RAND	2&3			14.5	100%	20.14	100%	17.71	93%
AV-PF-RAND	1&3	12.25	100%			34.62	60%		

Table5.11: Overall Results for Three Person Tracking for seq45-3p-1111. MAE is shown in terms of meters and the success rate of tracking is shown as percentage.

The speakers #1 and #3 are stable in Sequence #40. Hence, for stable targets, 2-D and

3-D trackers performances are similar. However, using two camera is beneficial for tracking of moving multi-targets as it can be concluded from results of the speaker #2 in Sequence #40.

Serious occlusion examples can be seen in Sequence #45. For these type of occlusions, the 3-D tracker fails. Since, the distorted measurements on one camera result to the failure of the overall tracker.

5.5.4 Conclusion

As it can be concluded from the presented results, by adding one camera to the 2-D tracker system, a multi-target tracking can be achieved in 3-D real world coordinate system. It is important, because that 3-D localization information can be used in different parts of the system for the variety of purposes.

Additionally, it can be concluded that integrating audio localization data to visual-only tracker which uses two cameras results no significant improvement on the tracking performance of the system. Integrating audio is useful, if there are serious visual occlusions in the scene.

CHAPTER 6

CONCLUSION

6.1 Conclusion

The main focus this thesis is on audio-visual fusion based on particle filter for multiple target tracking that works in indoor and noisy environments. Firstly, considering the tracking environment, AV16.3 dataset [23] is chosen to simulate the implemented methods. Since this dataset covers a variety of situations, e.g. visual occlusions and abrupt movements of the speakers. After that choice, two recent methods, namely V-PF and AV-PF, from the literature [18] are implemented for only the comparison. V-PF method is a particle filter which uses only visual cue to weight particles. In order to weight particles, the distance between the color histogram of the rectangular area around each particle is compared with the predefined image histogram of the target using Bhattacharyya distance measure. Second AV-PF method is a audio-visual particle filter which integrates the direction of arrival information to the location estimate of visual data. Both methods track the targets in the image plane of a camera.

In addition to those two methods, six different methods are implemented. For the visual part of the tracking, they shares the same histogram based approach. However, more simple and robust approach to integrate the audio and visual data are implemented in these new methods. With respect to that approach, in order to calculate the weight of a particle, two or more different weights can be used as the product of these weights for a given state. Although, one microphones array is used in the previous two methods, in new methods, two microphone arrays are used. Briefly, AV-PF-2D and AV-PF-2D-SNS are audio-visual particle filters with different audio integration

approaches and they also work on 2-D image planes. AV-PF-2D uses audio data also when the speaker is silent. However, AV-PF-2D-SNS discard the audio data for these periods of silence. Remaining four methods are implemented to track targets in 3-D real world coordinate system. V-PF-1CAM-3D uses only one camera and two microphone arrays to track target in 3-D world coordinate system. V-PF-2CAM uses only two cameras to track the targets. The implementation of this method aims to compare visual-only methods and audio-visual methods. AV-PF-3D uses two cameras and two microphone arrays for tracking in 3-D world coordinate system. Lastly, AV-PF-RAND adds random particle two 3-D real world coordinate system and ignores visual cues with low weights in order to prevent particle deprivation.

These eight methods can be classified as either 2-D or 3-D tracking methods with their tracking spaces. In the results, detailed analyses and the simulation results are presented for both type of methods. Also, by projecting the errors of 3-D methods to 2-D image plane, additional results are provided to compare 2-D and 3-D methods.

Consequently, it can be concluded that using two cameras is more effective than using only one camera. But, as expected, it needs more computational power. For 2-D cases, using only histogram based color cue is not enough for multi-target tracking. Furthermore, in 3-D cases, integrating audio cue to visual cue doesn't make a notable improvement on the accuracy and the robustness of the trackers. The main reason for that is the audio estimates our too noisy.

6.2 Future Works

The most obvious future work is to simulate implemented methods for different multi-target audio-visual datasets. With these implementations, the advantages and disadvantages of the methods can be analyzed in a more detailed way.

The audio estimates for the target localization in the implemented methods are too noisy. Only the direction estimates about the azimuth angles are reliable. Hence, by changing the geometry of the microphone arrays and using a 3-D configuration for microphones, a better results can be observed for the audio localization. This will also increase the performance the overall tracking system.

In the visual part, additional cues can be integrated. For example, texture cues and edge cues can be used. By integrating these cues to visual tracker part, the robustness of the tracker will probably increase in terms of success rate and the accuracy. However, this will also increase the need for computational power of the trackers.

Lastly, for a particle filter implementation, in order to optimize the computational cost, adaptive methods can be implemented. Implementation of the adaptive methods, especially for 3-D target tracking cases which contains relatively high number of particles, the computational efficiency of the tracking will probably increase.

Appendix A

GRAPHICAL RESULTS OF THE TRACKING METHODS

A.1 Graphical Results of 2-D Trackers

In this section, all the graphical results for 2-D trackers are presented. These graphs show the tracking error in terms of pixels for the related frame.

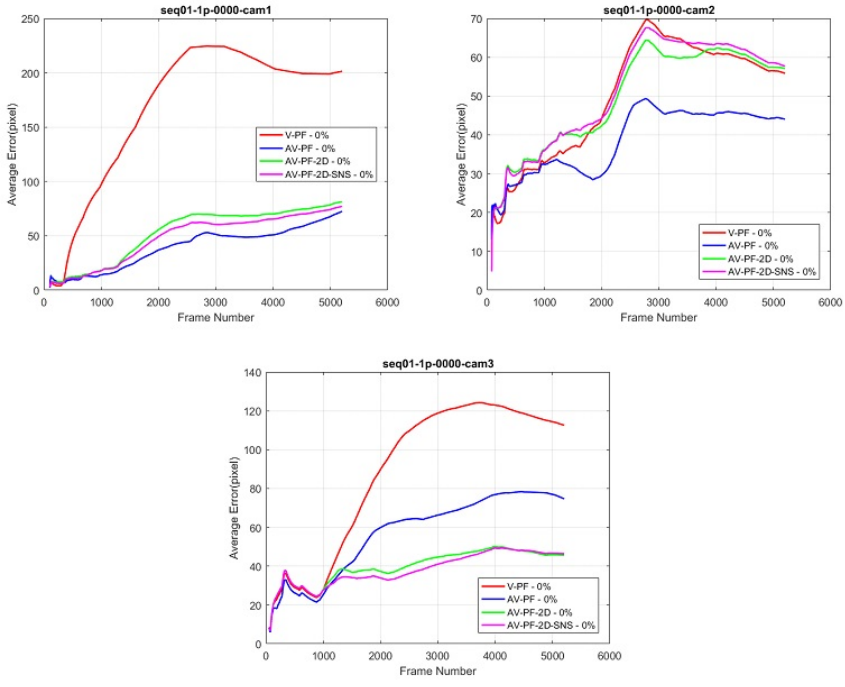


Figure A.1: Tracking Results of seq01-1p-0000 for 2-D Methods

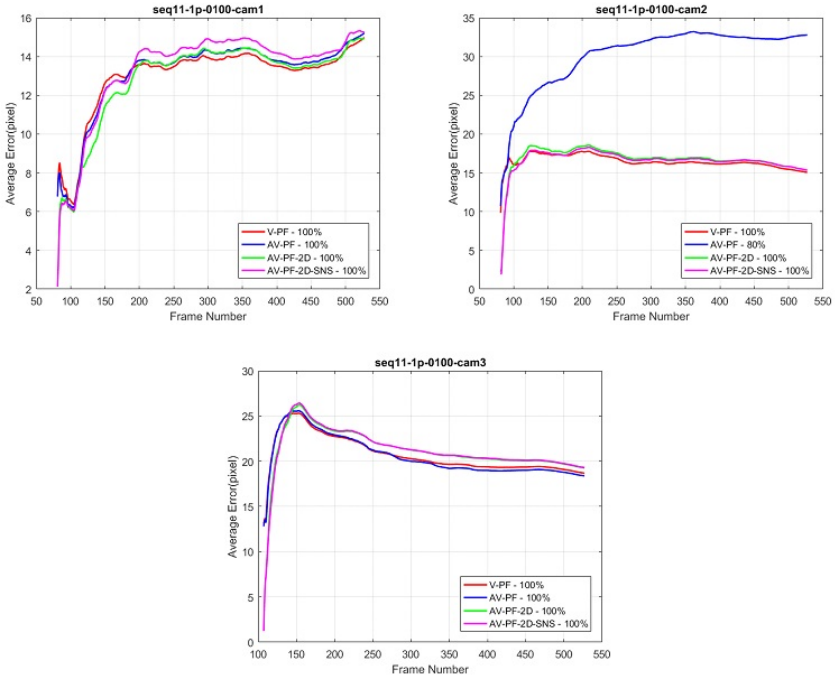


Figure A.2: Tracking Results of seq11-1p-0100 for 2-D Methods

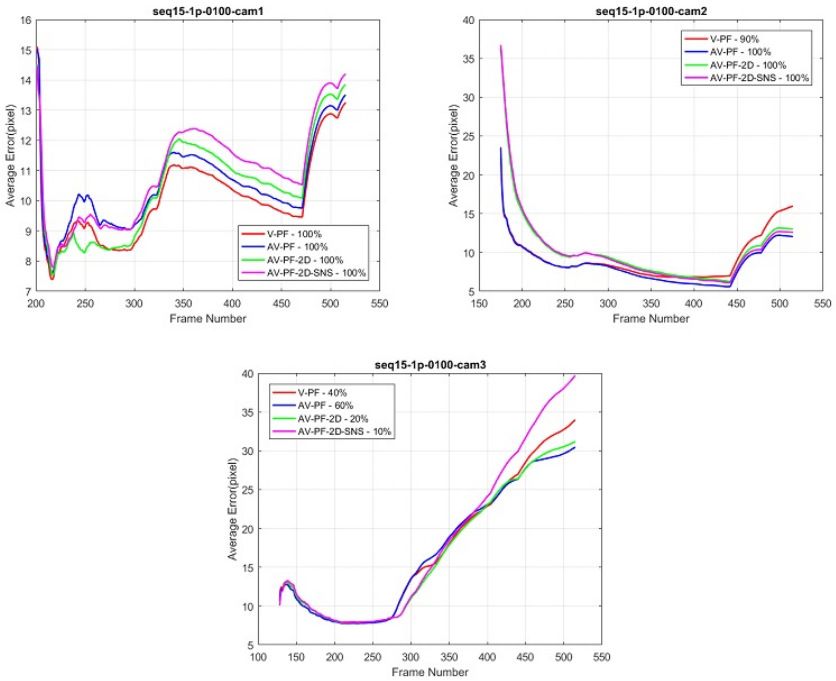


Figure A.3: Tracking Results of seq15-1p-0100 for 2-D Methods

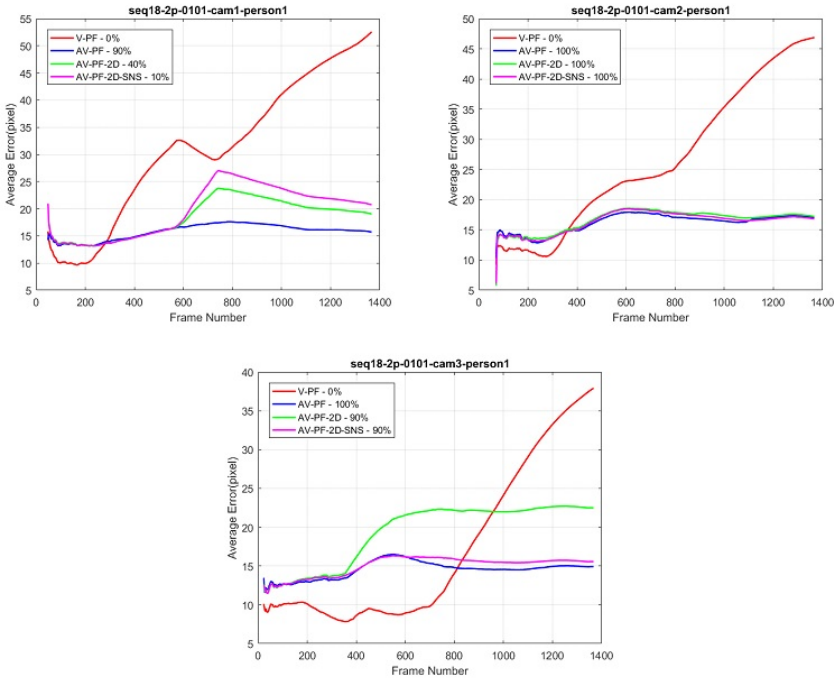


Figure A.4: Tracking Results of seq18-2p-0101 - Person #1 for 2-D Methods

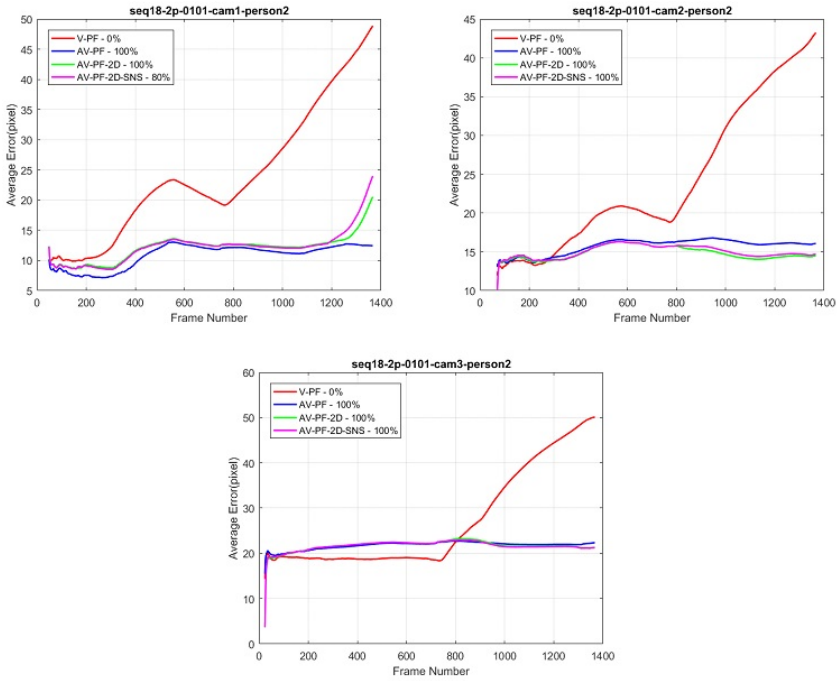


Figure A.5: Tracking Results of seq18-2p-0101 - Person #2 for 2-D Methods

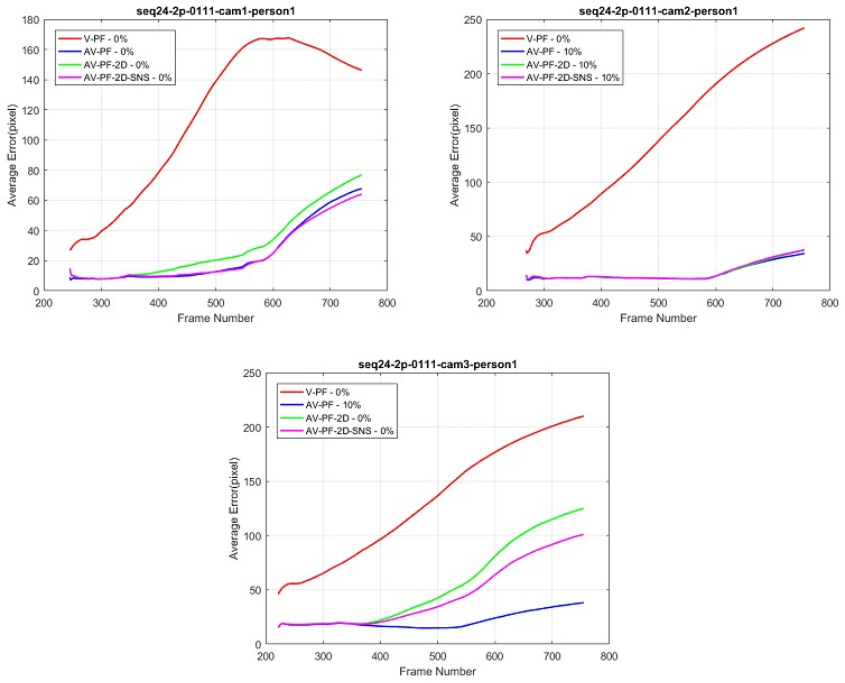


Figure A.6: Tracking Results of seq24-2p-0111 - Person #1 for 2-D Methods

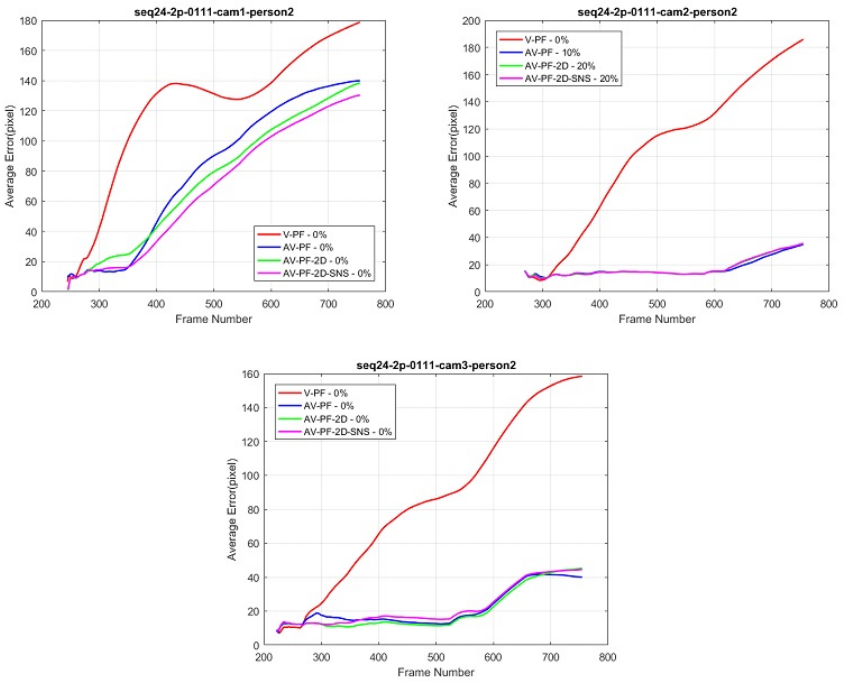


Figure A.7: Tracking Results of seq24-2p-0111 - Person #2 for 2-D Methods

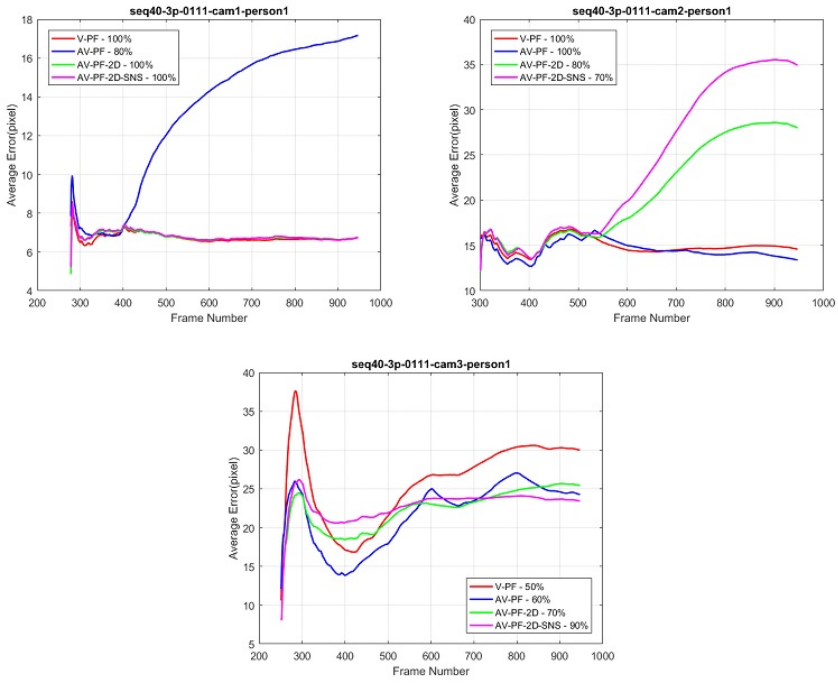


Figure A.8: Tracking Results of seq40-3p-0111 - Person #1 for 2-D Methods

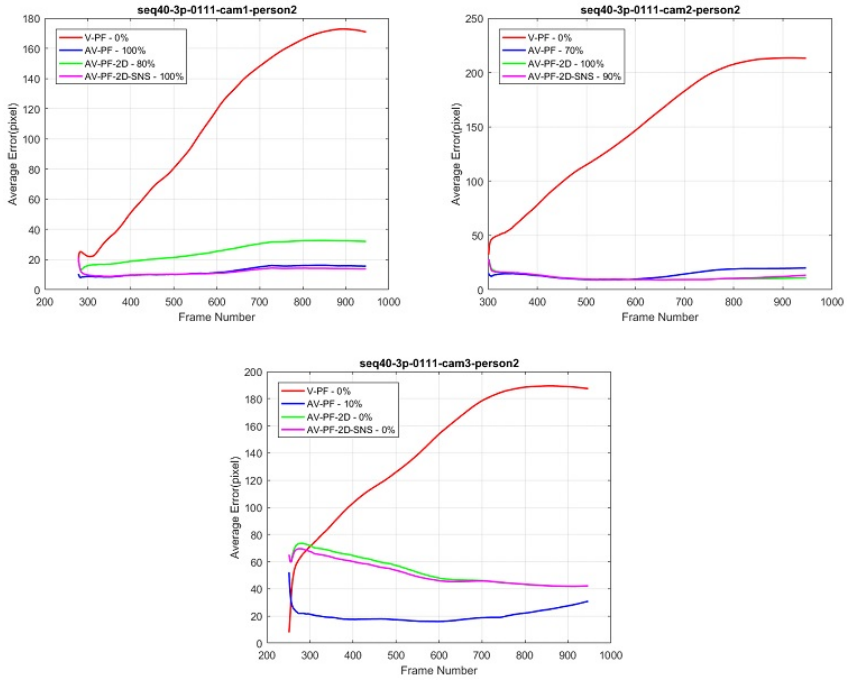


Figure A.9: Tracking Results of seq40-3p-0111 - Person #2 for 2-D Methods

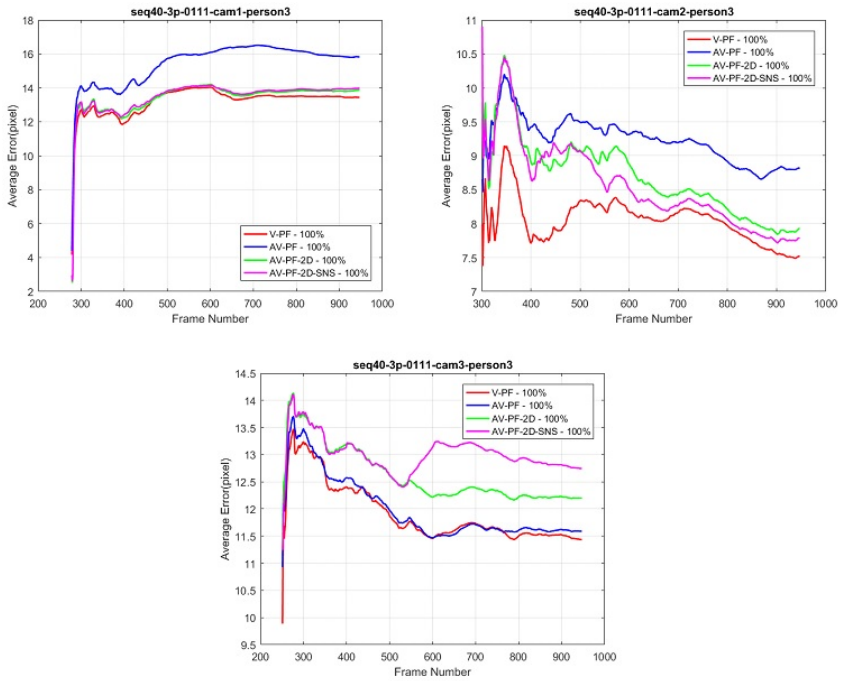


Figure A.10: Tracking Results of seq40-3p-0111 - Person #3 for 2-D Methods

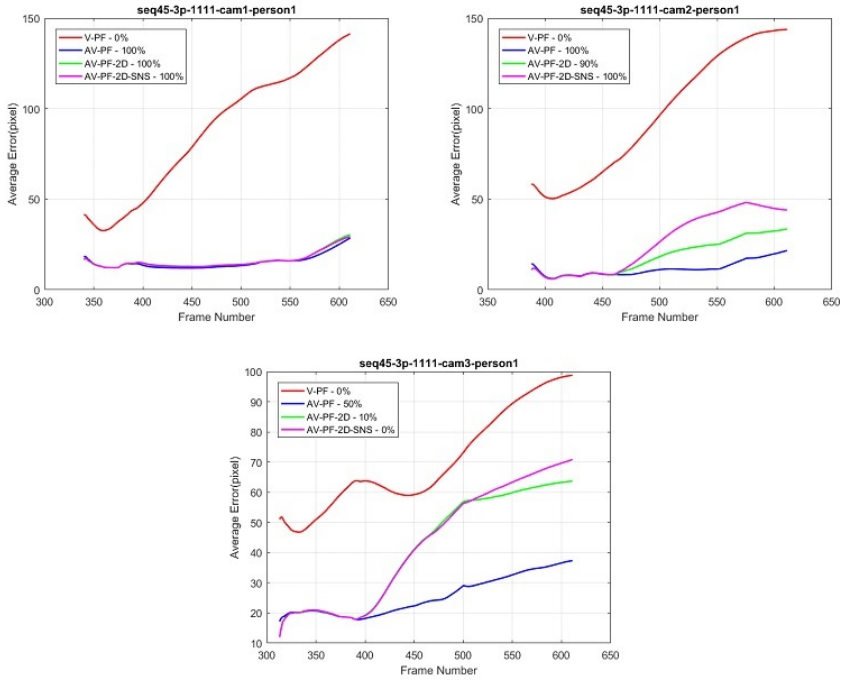


Figure A.11: Tracking Results of seq45-3p-1111 - Person #1 for 2-D Methods

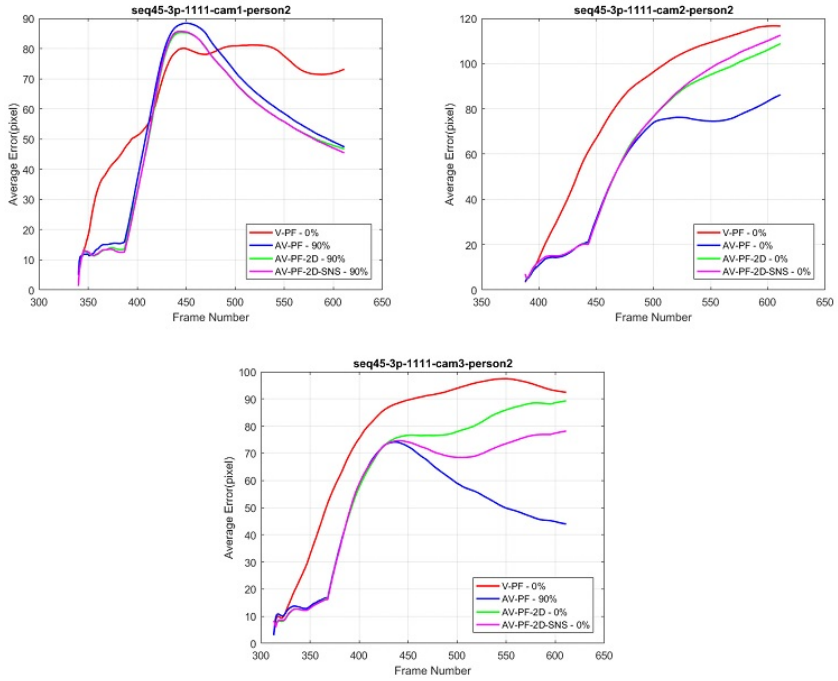


Figure A.12: Tracking Results of seq45-3p-1111 - Person #2 for 2-D Methods

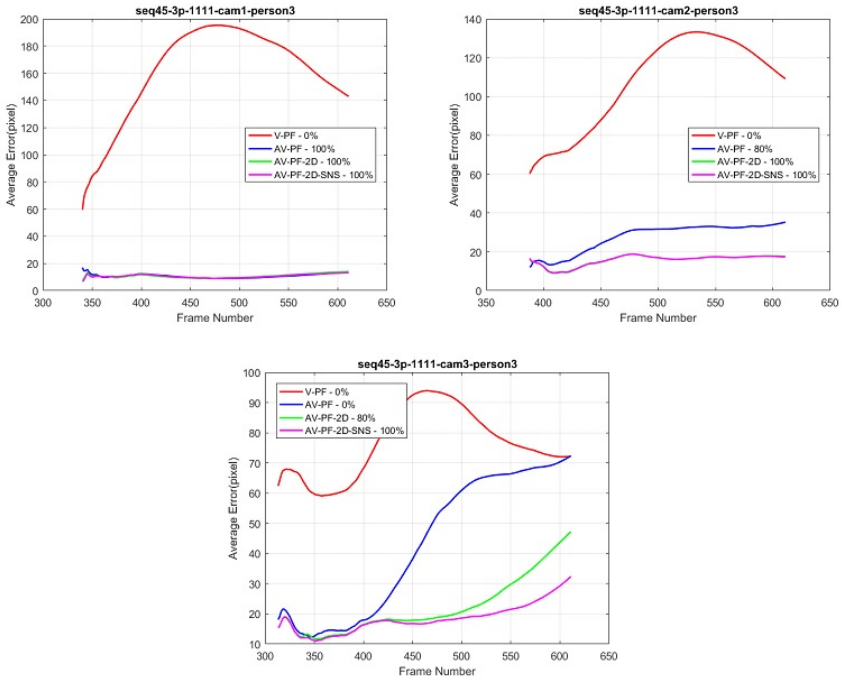


Figure A.13: Tracking Results of seq45-3p-1111 - Person #3 for 2-D Methods

A.2 Graphical Results of 3-D Trackers

In this section, all the graphical results for 3-D trackers are presented. These graphs show the tracking error in terms of meters for the related frame.

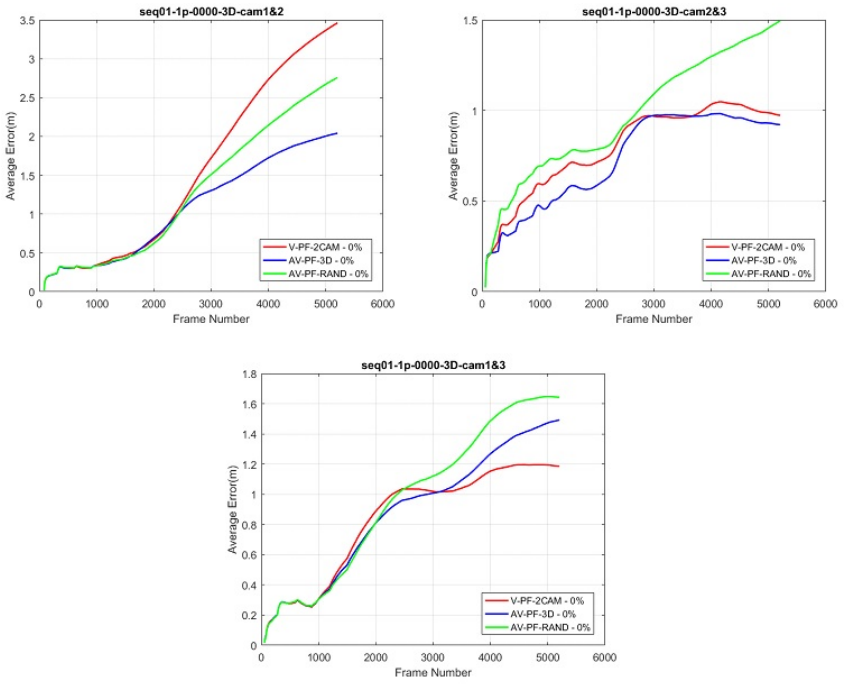


Figure A.14: Tracking Results of seq01-1p-0000 for 3-D Methods

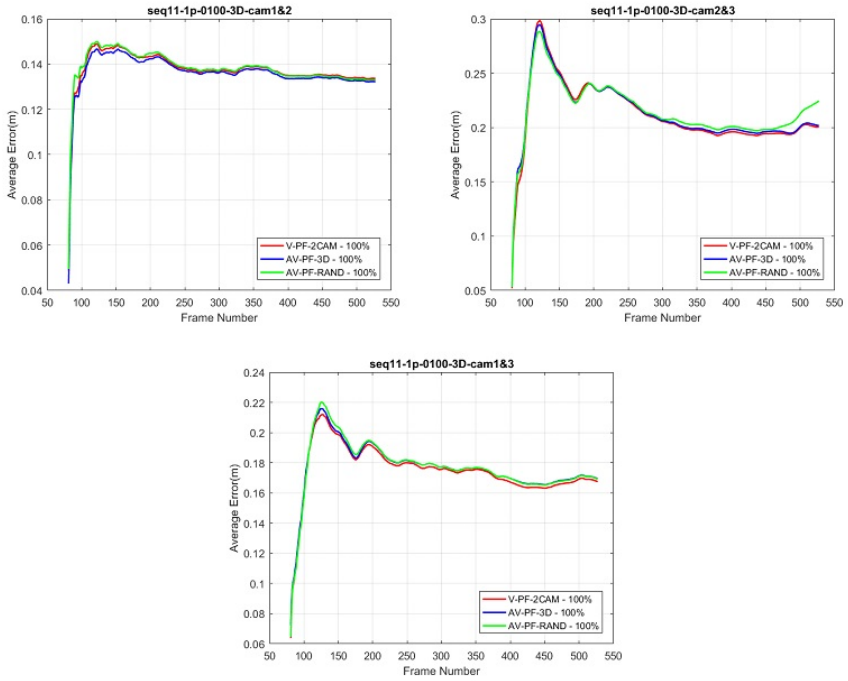


Figure A.15: Tracking Results of seq11-1p-0100 for 3-D Methods

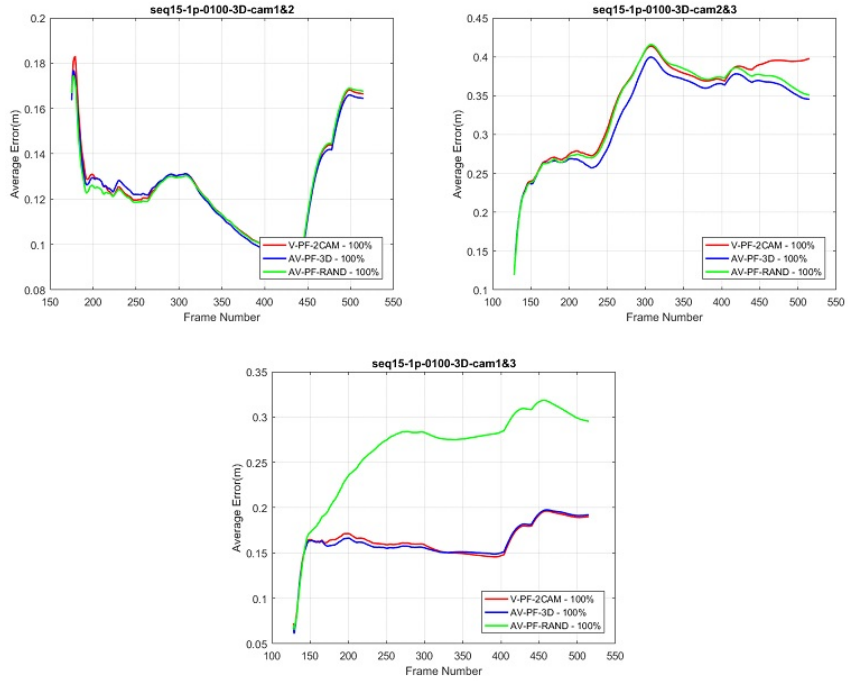


Figure A.16: Tracking Results of seq15-1p-0100 for 3-D Methods

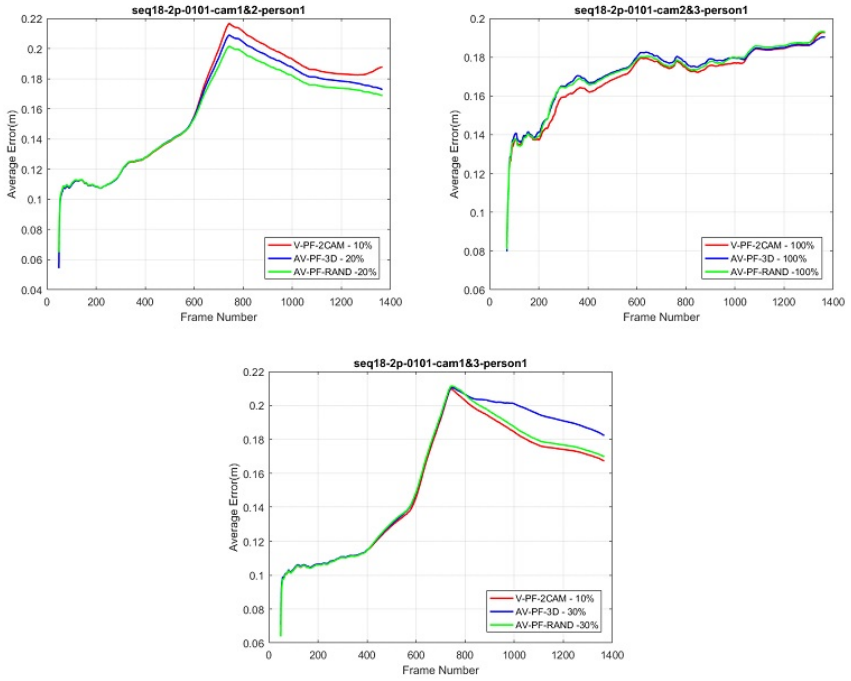


Figure A.17: Tracking Results of seq18-2p-0101 - Person #1 in 3-D

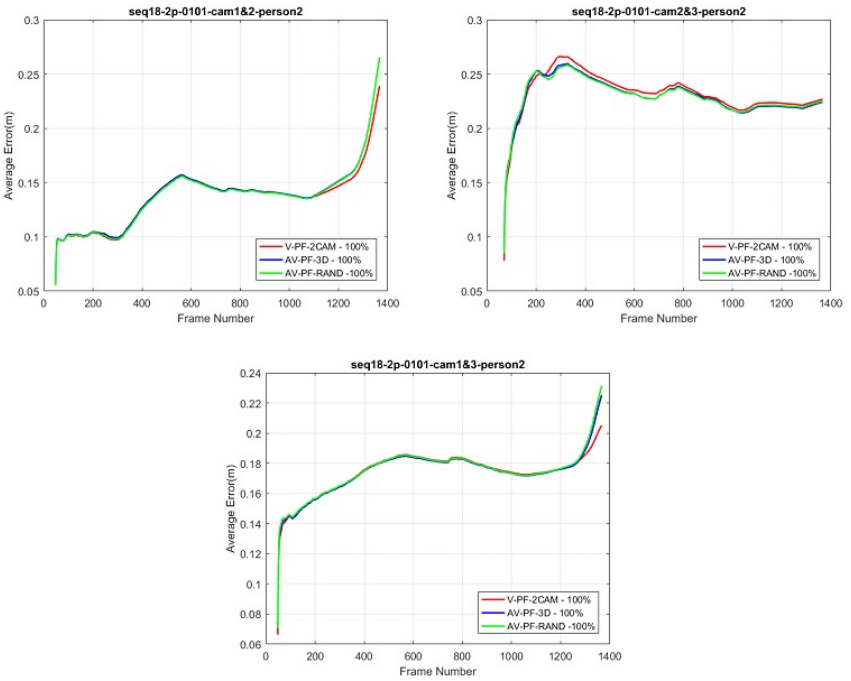


Figure A.18: Tracking Results of seq18-2p-0101 - Person #2 in 3-D

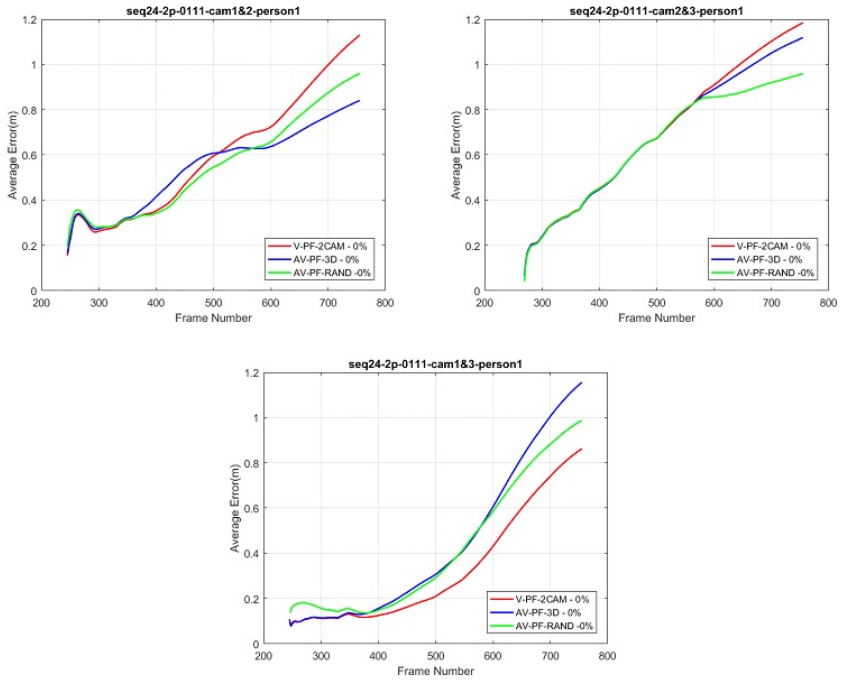


Figure A.19: Tracking Results of seq24-2p-0111 - Person #1 in 3-D

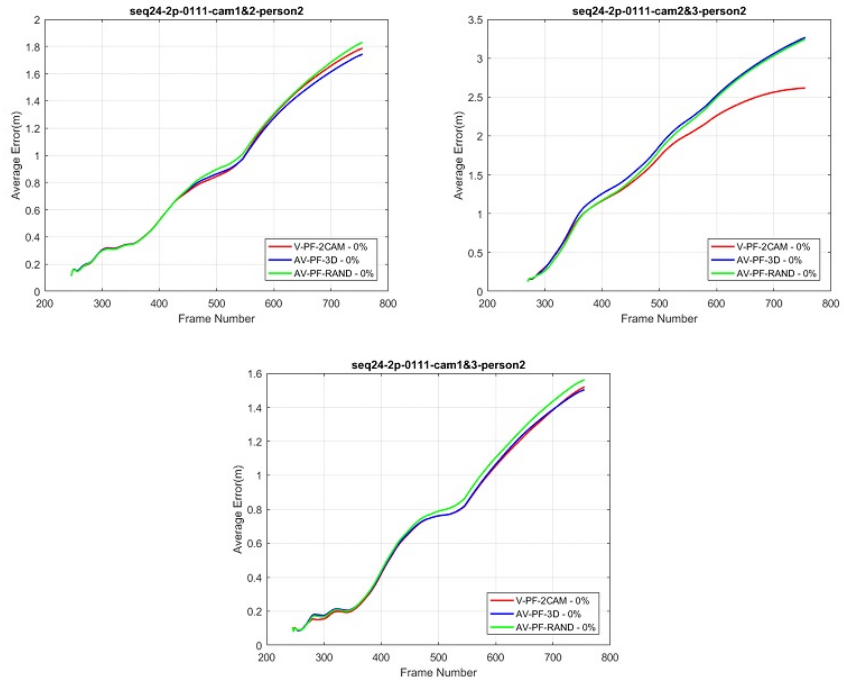


Figure A.20: Tracking Results of seq24-2p-0111 - Person #2 in 3-D

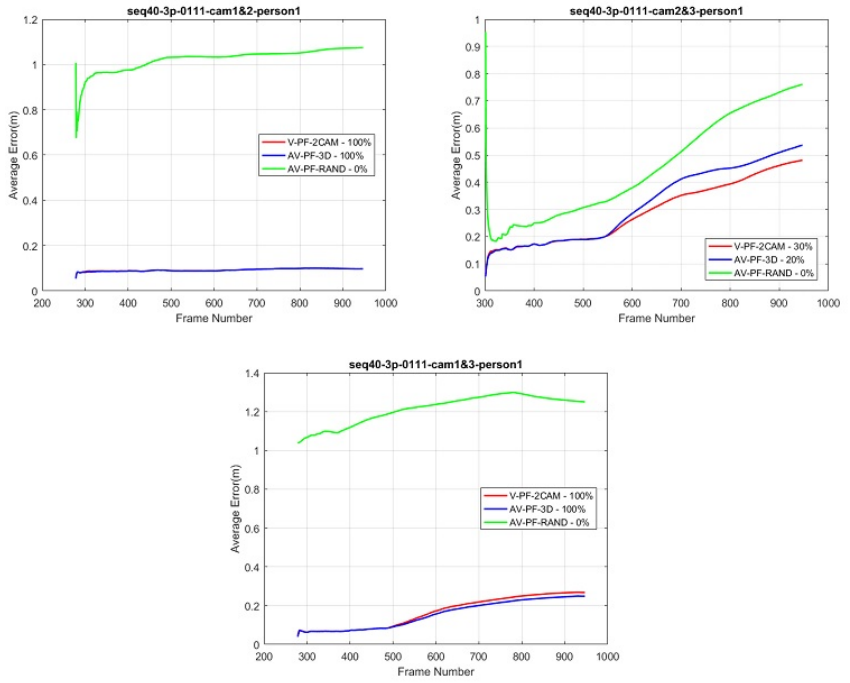


Figure A.21: Tracking Results of seq40-3p-0111 - Person #1 in 3-D

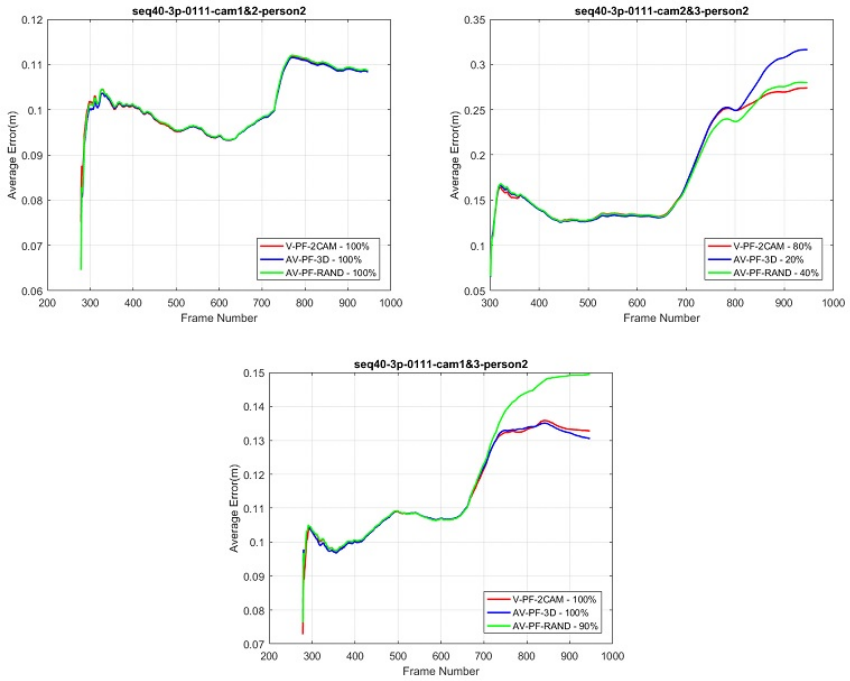


Figure A.22: Tracking Results of seq40-3p-0111 - Person #2 in 3-D

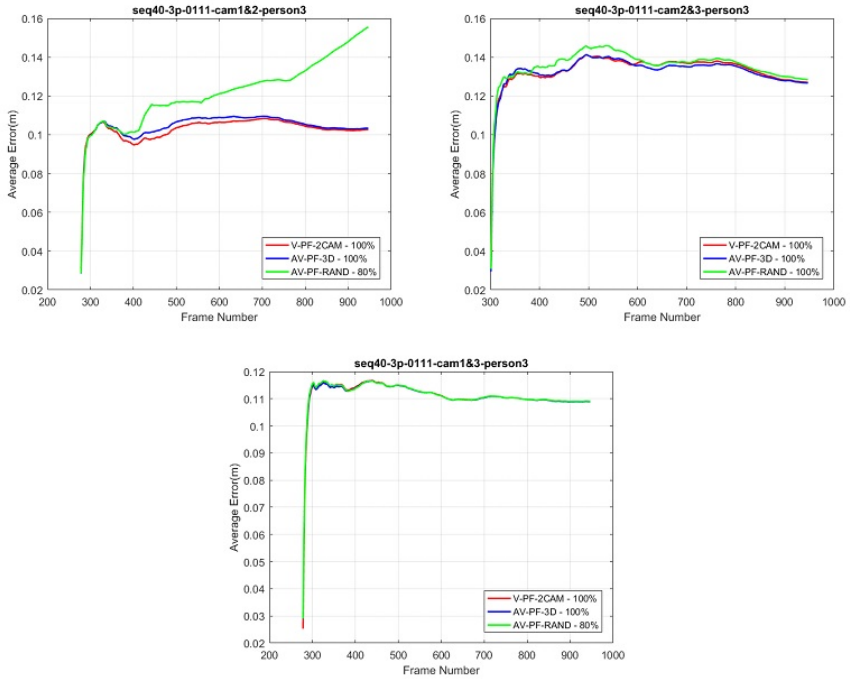


Figure A.23: Tracking Results of seq40-3p-0111 - Person #3 in 3-D

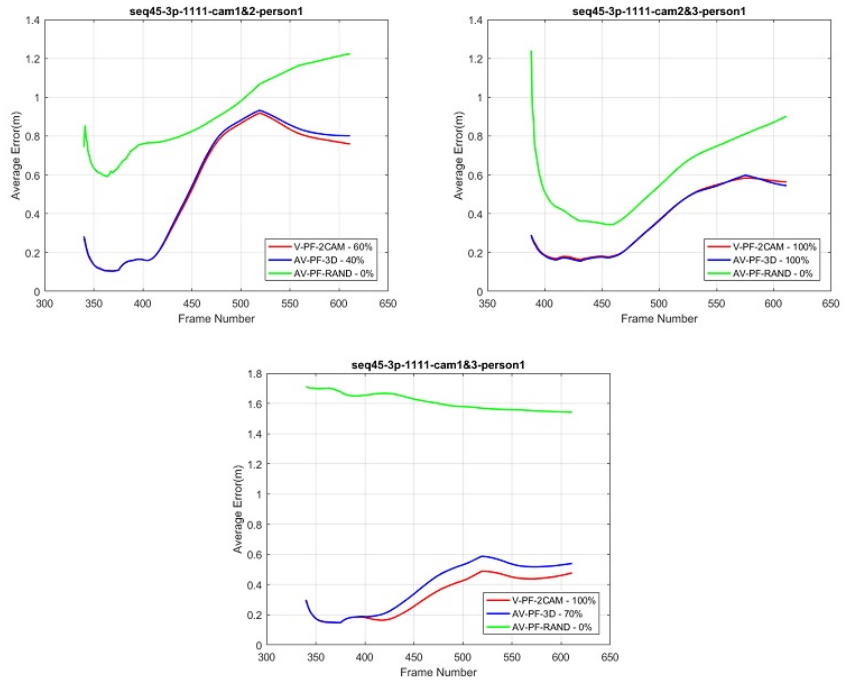


Figure A.24: Tracking Results of seq45-3p-1111 - Person #1 in 3-D

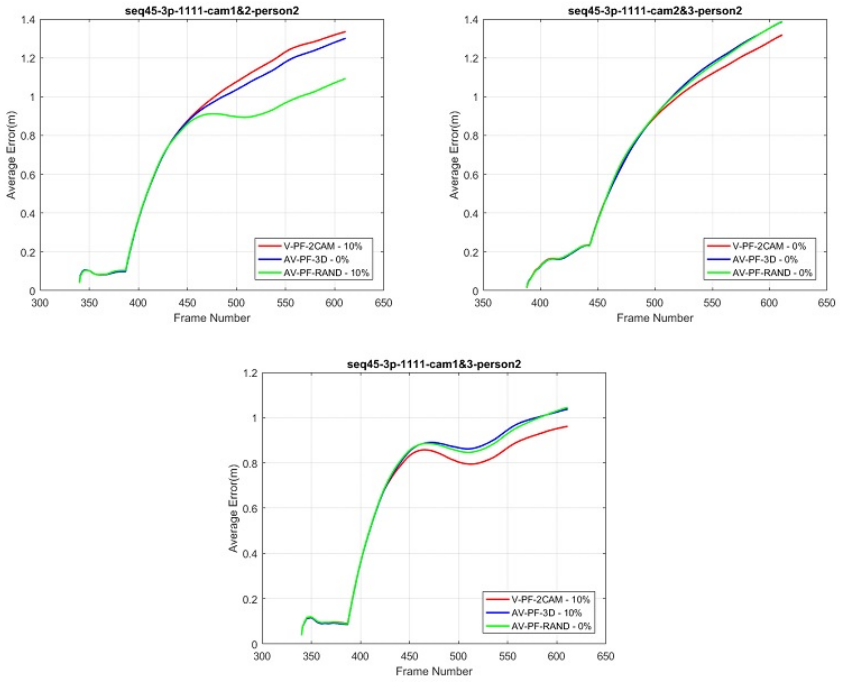


Figure A.25: Tracking Results of seq45-3p-1111 - Person #2 in 3-D

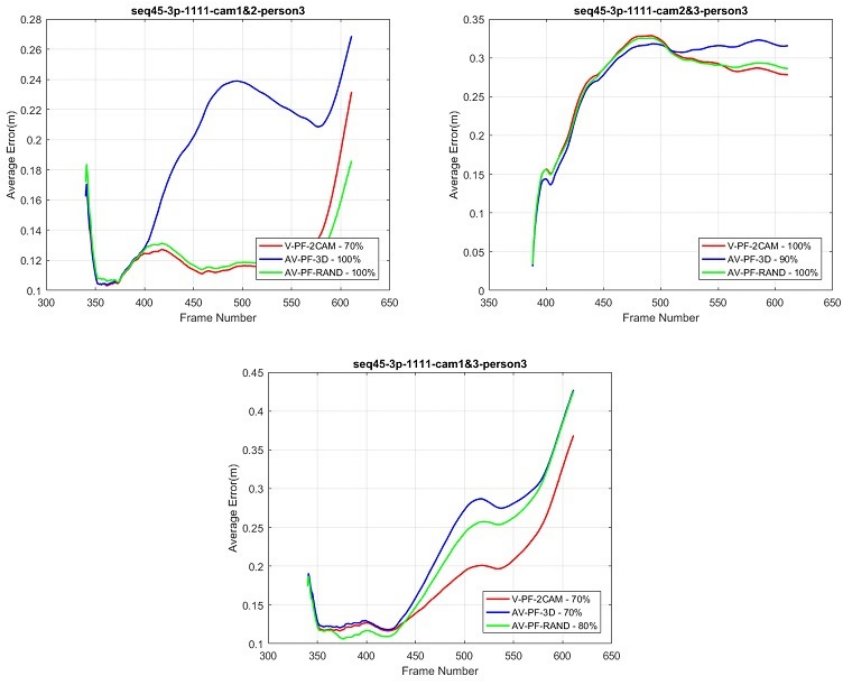


Figure A.26: Tracking Results of seq45-3p-1111 - Person #3 in 3-D

REFERENCES

[1] H. Agirman, S. Karakutuk, and E. Gonendik. Design and implementation of a microphone array system with flexible array geometry [degisebilir geometrili mikrofon dizin sistemi tasarim ve uygulamasi]. *2014 22nd Signal Processing and Communications Applications Conference, SIU 2014 - Proceedings*, pages 2054–2057, 2014.

[2] P. K. Atrey, M. A. Hossain, A. E. Saddik, and M. S. Kankanhalli. Multimodal fusion for multimedia analysis: A survey. *Multimedia systems*, 16(6):345–379, 2010.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[4] M. J. Beal, N. Jojic, and H. Attias. A graphical model for audiovisual object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(7):828–836, 2003.

[5] K. Bernardin, T. Gehrig, and R. Stiefelhagen. Multi-level particle filter fusion of features and cues for audio-visual person tracking. In *Multimodal Technologies for Perception of Humans*, pages 70–81. Springer, 2008.

[6] W. D. Blair. Design of nearly constant velocity track filters for brief maneuvers. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8. IEEE, 2011.

[7] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah. Sequential Monte Carlo tracking by fusing multiple cues in video sequences. *Image and Vision Computing*, 25(8):1217–1227, 2007.

[8] D. C. Brown. Decentering distortion of lenses. *Photometric Engineering*, 32(3):444–462, 1966.

[9] A. L. Casanovas. *Audio-Visual Fusion: New Methods and Applications*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2011.

[10] B. Cyganek and J. P. Siebert. *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, 2011.

[11] D. A. Forsyth and J. Ponce. A modern approach. *Computer Vision: A Modern Approach*, pages 88–101, 2003.

[12] D. Gatica-Perez, G. Lathoud, I. A. McCowan, J.-M. Odobez, and D. Moore. Audio-visual speaker tracking with importance particle filters. Technical report, IDIAP, 2002.

[13] D. Gatica-Perez, G. Lathoud, J.-M. Odobez, and I. McCowan. Audiovisual probabilistic tracking of multiple speakers in meetings. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(2):601–616, 2007.

[14] T. Gehrig, K. Nickel, H. K. Ekenel, U. Klee, and J. McDonough. Kalman filters for audio-video source localization. In *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*, pages 118–121. IEEE, 2005.

[15] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. 140(2):107–113, 1993.

[16] J. M. Hammersley and K. W. Morton. Poor man's monte carlo. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 23–38, 1954.

[17] A. K. Katsaggelos, S. Bahaadini, and R. Molina. Audiovisual fusion: Challenges and new approaches. *Proceedings of the IEEE*, 103(9):1635–1653, September 2015.

[18] V. Kılıç, M. Barnard, W. Wang, and J. Kittler. Audio assisted robust visual tracking with adaptive particle filtering. *IEEE Transactions on Multimedia*, 17(2):186–200, February 2015.

[19] M. Krupa. Verification of particle filtering based framework implemented in MATLAB®. *Seek Digital Library*, 2012.

[20] O. Lanz. Approximate Bayesian multibody tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1436–1449, 2006.

[21] G. Lathoud. *Spatio-Temporal Analysis of Spontaneous Speech with Microphone Arrays*. PhD thesis, École Polytechnique Fédérale de Lausanne, December 2006.

[22] G. Lathoud and M. Magimai-Doss. A sector-based, frequency-domain approach to detection and localization of multiple speakers. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 3:iii–265, 2005.

[23] G. Lathoud, J.-M. Odobez, and D. Gatica-Perez. AV16.3: an audio-visual corpus for speaker localization and tracking. *Machine Learning for Multimodal Interaction*, pages 182–195, 2005.

[24] A. P. Loh, F. Guan, and S. S. Ge. Motion estimation using audio and video fusion. In *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, volume 3, pages 1569–1574. IEEE, 2004.

[25] R. C. Luo and M. G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Trans. Syst. Man Cybern.*, 19(5):901–931, September/October 1989.

[26] D. Moore. The IDIAP smart meeting room. Technical report, IDIAP, 2002.

[27] K. Nickel, T. Gehrig, R. Stiefelhagen, and J. McDonough. A joint particle filter for audio-visual speaker tracking. In *Proceedings of the 7th international conference on Multimodal interfaces*, pages 61–68. ACM, 2005.

[28] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and vision computing*, 21(1):99–110, 2003.

[29] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Computer vision—ECCV 2002*, pages 661–675. Springer, 2002.

[30] S. T. Shivappa, M. M. Trivedi, and B. D. Rao. Audiovisual information fusion in human-computer interfaces and intelligent environments: A survey. *Proceedings of the IEEE*, 98(10):1692–1715, 2010.

[31] F. Talantzis, A. Pnevmatikakis, and L. C. Polymenakos. Real time audio-visual person tracking. In *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, pages 243–247. IEEE, 2006.

[32] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[33] L. Turner and C. Sherlock. An introduction to particle filtering. 2013.

[34] J. Vermaak, M. Gangnet, A. Blake, and P. Perez. Sequential Monte Carlo fusion of sound and vision for speaker tracking. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 741–746. IEEE, 2001.

[35] Y. Wang, Z. Liu, and J.-C. Huang. Multimedia content analysis-using both audio and visual clues. *Signal Processing Magazine, IEEE*, 17(6):12–36, 2000.

[36] G. Yunhai, W. Shuang, and C. Binglong. Calibration for star tracker with lens distortion. In *Mechatronics and Automation (ICMA), 2012 International Conference on*, pages 681–686. IEEE, 2012.

[37] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673. IEEE, 1999.

[38] D. N. Zotkin, R. Duraiswami, and L. S. Davis. Joint audio-visual tracking using particle filters. *EURASIP Journal on Applied Signal Processing*, 2002(1):1154–1164, 2002.