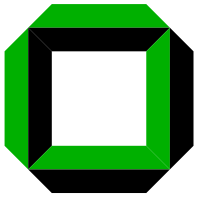

Universität Karlsruhe (TH)
Institut für
Theoretische Informatik

Interactive Systems Laboratories

Kalman Filters for Audio-Video Source Localization

Tobias Gehrig

Studienarbeit

Verantwortlicher Betreuer (Prof.): Prof. Alex Waibel
Betreuende Mitarbeiter: Dr. John McDonough

26. Februar 2007

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfaßt und keine anderen als die angegebenen Literaturhilfsmittel verwendet zu haben.

I hereby declare that this thesis is a work of my own, and that only cited sources have been used.

Karlsruhe, den 26. Februar 2007

Tobias Gehrig

Abstract

In this thesis an approach is presented to do speaker tracking using audio-visual features, namely time delay of arrival estimation on microphone array signals and face detection on multiple camera images. The features are then used as input for an iterated extended Kalman filter for the actual speaker tracking. This is done without any closed-form approximations, instead the Kalman filter does the sensor fusion. The approach was evaluated on real lecture data to provide experimental results that show how this technique performs in a real-life scenario. These show that the proposed speaker tracker performs better when using both audio and video features than using the audio-only or video-only features.

Contents

Chapter 1

Introduction

For far-field speech recognition microphone arrays are used to record the speaker's voice [WNM05]. Since far-field microphones record more than just the desired speaker we need a way to improve the signal quality of those far-field microphones. This can be done by doing *beamforming* [Van02]. In the simplest case, the signals recorded by each microphone are delayed such that the desired signal is in sync for all channels and summed up afterward. This is called *delay-and-sum beamforming*. But for this to work we have to know the position of the speaker. Here is where the approach presented in this thesis comes in. It localizes the person who is currently speaking and afterwards the position can be used by the beamformer to improve signal quality. Another application is in face recognition, where a high resolution image of the face of the person to be recognized is needed. Thus a pan-tilt-zoom camera is steered at the estimated position of the speaker so that a face recognizer can then identify the speaker using the close-up image [BES06]. This identification can then further be used for transcription of dialogs in a meeting.

The speaker tracking algorithm presented in this thesis uses audio-visual features. As the audio feature the *time delays of arrival* (TDOA) [KC76] estimated from the signal of some microphone arrays is used. The two-dimensional position estimate returned from a face detector running on the video images of some cameras serves as video feature [JV03].

In previous approaches where only audio features were used [KGMon] there have been problems in tracking the speaker when he was not speaking or some noise source happened to be louder and more dominant than the speaker's voice. On the other hand, video only trackers have problems in selecting the speaker out of the many visual objects and keeping track of the right object when it happens to be occluded or otherwise undetectable at the moment.

To overcome the problems that occur when using only one of those two features our approach uses both features together. This is done by updating a *Kalman filter* sequentially with audio features and video features much like described in [Wel96, WB97], instead of updating the Kalman filter with explicit position estimates made by the individual sensors [SSR01].

Another thing is that audio source localization is often done by using some closed-form approximations for TDOA estimation [CH94, SA87]. On the other hand, when using multiple cameras for video source localization, there is a need to get from the two-dimensional points detected in the individual views to a three-dimensional position estimate. This is often addressed by doing triangulation [FS02]. But here this is all

integrated into the Kalman filter similar to [DZD01].

In the following chapters we will look into those features more deeply. In chapter 2 the time delay of arrival and its estimation for the audio features will be presented, followed by the projection and face detection for the video features in chapter 3. Afterwards, in the 4th chapter, there will be a section about the speaker tracking with the underlying concept of Kalman filters and some refinements that improve the accuracy and speed of the tracker. Then the setup and the results of the experiments are presented and discussed in chapter 5 and finally there will be conclusions about this topic in chapter 6.

Chapter 2

Audio Features

In a seminar scenario it is of interest to track the speaker, who is currently speaking, so that beamforming can be done about the estimated position to get a cleaner signal for speech recognizers. Or a point-to-zoom camera can be pointed at that person to be able to do face identification or simply a close-up of the speaker's face.

2.1 Time Delay of Arrival

If we record the speaker using two microphones, the speaker's voice arrives at the two microphones at different times, so we can take the difference between the times of arrival as a feature for inferring the speaker's position. This difference is called the *time delay of arrival* (TDOA) (see Figure 2.1). To do a more accurate position estimation we use more than one pair of microphones.

So consider the i-th pair of microphones, and let $\mathbf{m_{i1}}$ and $\mathbf{m_{i2}}$ be the 3-dimensional positions of the first and second microphone in this pair and $\mathbf{x}$ the 3-dimensional position of the speaker. To express the time delay of arrival we need to calculate the distance between the speaker and each microphone and divide it by the speed of sound s this leads to the following equation:

$$T_i(\mathbf{x}) = T(\mathbf{m_{i1}}, \mathbf{m_{i2}}, \mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{m_{i1}}\| - \|\mathbf{x} - \mathbf{m_{i2}}\|}{s} \tag{2.1}$$

which we can simplify by substituting the distance to

$$\begin{aligned} d_{ij} &= \sqrt{(x - m_{ij,x})^2 + (y - m_{ij,y})^2 + (z - m_{ij,z})^2} \\ &= \|\mathbf{x} - \mathbf{m}_{ij}\| \end{aligned} \tag{2.2}$$

resulting in

$$T_i(\mathbf{x}) = \frac{1}{s}(d_{i1} - d_{i2}) \tag{2.3}$$

This formula gives us the theoretical TDOA for a given speaker position, which can then be compared to a measured TDOA.

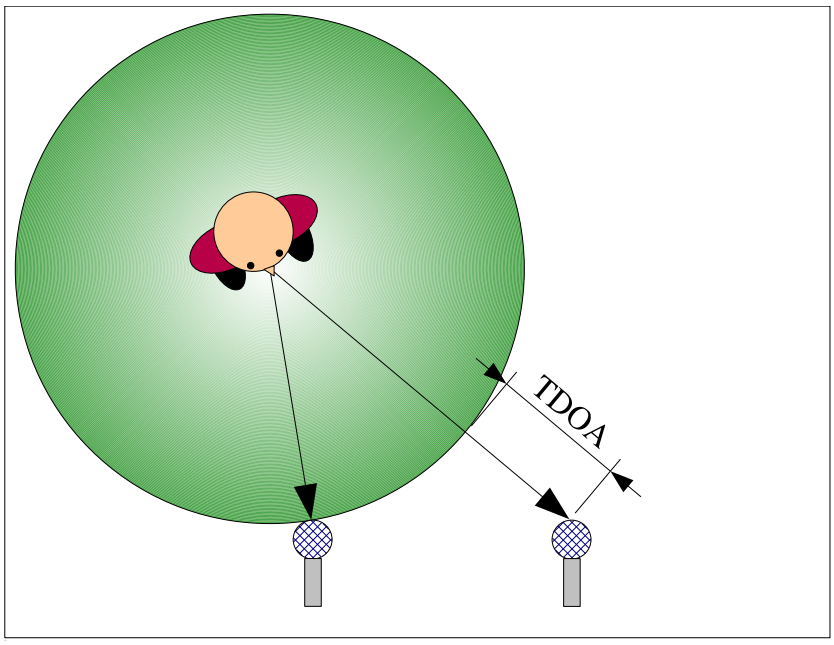

Figure 2.1: The speech of the speaker arrives at the microphones at different times resulting in a time delay of arrival (TDOA).

2.2 Time Delay of Arrival Estimation

This measurement can be done by using the generalized cross correlation (GCC) [KC76], that is expressed as

$$R_{12}(\tau) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(e^{j\omega\tau}) X_1(e^{j\omega\tau}) X_2^*(e^{j\omega\tau})\, e^{j\omega\tau}\, d\omega \tag{2.4}$$

where $W(e^{j\omega\tau})$ is a frequency dependent weight that is multiplied with the product of the two frequency components $X_1(e^{j\omega\tau})$ and $X_2(e^{j\omega\tau})$ of the two microphones and afterwards integrated over the whole frequency spectrum. In the most common case this weight is given by

$$W_{PHAT}(e^{j\omega\tau}) = \frac{1}{|X_1(e^{j\omega\tau}) X_2^*(e^{j\omega\tau})|} \tag{2.5}$$

This variant of the GCC is called the phase transform (PHAT) [OS94, KC76], because it normalizes the the amplitude of all frequencies and leaves only the phase for the calculation of the cross correlation.
For efficient computation $R_{12}(\tau)$ is calculated with an inverse FFT.
To get the TDOA we search for the maximum in the resulting cross correlation function and have the TDOA defined as the position of that maximum in the GCC:

$$\hat{\tau}_i = \arg\max_{\tau} R_{12}(\tau) \tag{2.6}$$

Now that we have the theoretical TDOA $T_i(\mathbf{x})$ and observed TDOA $\hat{\tau}_i$ we can minimize the error function

$$\epsilon(\mathbf{x}) = \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \left[\hat{\tau}_i - T_i(\mathbf{x})\right]^2 \tag{2.7}$$

that gives us source localization based on a maximum likelihood (ML) criterion [Kay93], where σ_i^2 denotes the error covariance of this observation. Therefore we need to solve for the position $\mathbf{x}$ minimizing (2.7), but since (2.1) is nonlinear in $\mathbf{x}$ we will find it useful to have a linearization of this function.
So let us approximate $T_i(\mathbf{x})$ with a first order Taylor series expansion about the last position estimate $\hat{x}(t-1)$, even though (2.7) implies finding the $\mathbf{x}$ which minimizes the instantaneous error criterion rather than minimizing over a series of time instants. But since the speakers's position cannot change instantaneously, both the present $\hat{\tau}_i(t)$ and past TDOA estimates $\{\hat{\tau}_i(j)\}_{j=0}^{t-1}$ are useful for the position estimation. This gives us the following equation:

$$T_i(\mathbf{x}) \approx T_i(\hat{\mathbf{x}}(t-1)) + \mathbf{c}_i^T(t)[\mathbf{x} - \hat{\mathbf{x}}(t-1)] \tag{2.8}$$

where

$$\mathbf{c}_i(t) = \left[\nabla_{\mathbf{x}} T_i(\mathbf{x})\right]_{\mathbf{x}=\hat{\mathbf{x}}(t-1)} \tag{2.9}$$

Now we need the gradient of the TDOA function to complete the linearization. So let us first take the partial derivative of $T_i(\mathbf{x})$ with respect to x:

$$\frac{\delta T_i(\mathbf{x})}{\delta x} = \frac{1}{s} \cdot \left[\frac{x - m_{i1,x}}{d_{i1}} - \frac{x - m_{i2,x}}{d_{i2}}\right] \tag{2.10}$$

Seeing that the partial derivatives with respect to y and z are anologues, we can write the gradient as follows:

$$\nabla_{\mathbf{x}} T_i(\mathbf{x}) = \frac{1}{s} \cdot \left[\frac{\mathbf{x} - \mathbf{m}_{i1}}{d_{i1}} - \frac{\mathbf{x} - \mathbf{m}_{i2}}{d_{i2}}\right] \tag{2.11}$$

Substituting (2.11) into (2.9) leads to

$$\mathbf{c}_i(t) = \left[\nabla_{\mathbf{x}} T_i(\mathbf{x})\right]_{\mathbf{x}=\hat{\mathbf{x}}(t-1)} = \frac{1}{s} \cdot \left[\frac{\mathbf{x} - \mathbf{m}_{i1}}{d_{i1}} - \frac{\mathbf{x} - \mathbf{m}_{i2}}{d_{i2}}\right]_{\mathbf{x}=\hat{\mathbf{x}}(t-1)} \tag{2.12}$$

In Figure 2.2 we can see how well the linearization performs in comparison to the nonlinear TDOA estimation. The reference point used for the linearization is approximately the center of the room $\begin{bmatrix}2950 & 4080 & 1700\end{bmatrix}$. As we can see the error within 1 m about the estimated point is modest. The RMS error for the two figures is $2.765 \cdot 10^{-6}$ s for movement in x direction and $1.031 \cdot 10^{-6}$ s for movement in y direction. The worst-case deviation is 2.33 % in x and 1.38 % in y. So assuming that we have a accuracy of 1 m to localize the speaker the linearization performs almost as good as the nonlinear TDOA estimation.

Now that we have a linearization of the TDOA function we can rewrite the error function by substituting (2.8) into (2.7):

$$\begin{aligned}\epsilon(\mathbf{x};t) &\approx \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \{\hat{\tau}_i - T_i(\hat{\mathbf{x}}(t-1)) - \mathbf{c}_i(t)^T[\mathbf{x} - \hat{\mathbf{x}}(t-1)]\}^2 \\ &= \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} [\bar{\tau}_i - \mathbf{c}_i(t)^T \mathbf{x}]^2 \end{aligned} \tag{2.13}$$

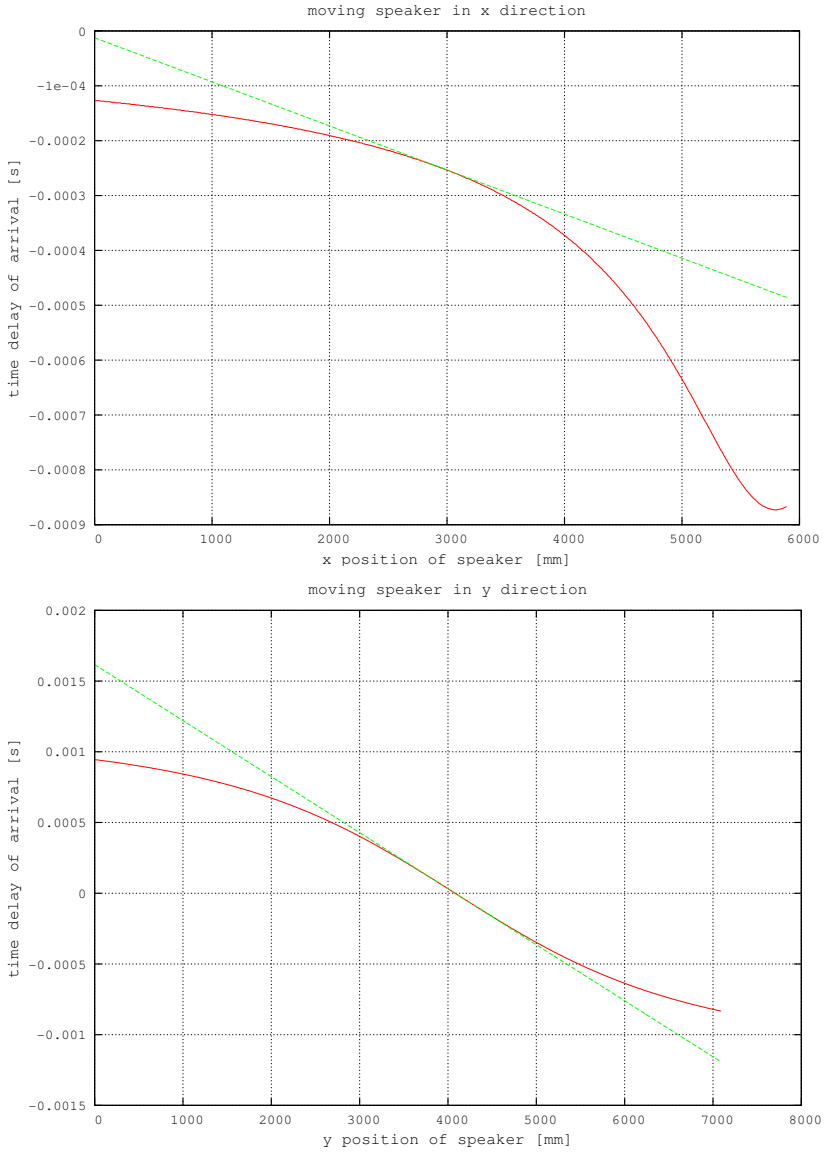

Figure 2.2: Comparison between nonlinear function and first order Taylor series. On the top the microphones used are on top of each other and the speaker is moving in the x direction. On the bottom the microphones are beside each other and the speaker is moving in the y direction.

where

$$\bar{\tau}_i(t) = \hat{\tau}_i(t) - T_i(\hat{\mathbf{x}}(t-1)) + \mathbf{c}_i^T(t)\hat{\mathbf{x}}(t-1) \tag{2.14}$$

for $i = 0, \ldots, N-1$. Since we have multiple microphone pairs, we want to have a formula that deals with all microphones pairs at once. Let us define

$$\bar{\boldsymbol{\tau}} = \begin{pmatrix} \bar{\tau}_0(t) \\ \bar{\tau}_1(t) \\ \vdots \\ \bar{\tau}_{N-1}(t) \end{pmatrix} \qquad \hat{\boldsymbol{\tau}} = \begin{pmatrix} \hat{\tau}_0(t) \\ \hat{\tau}_1(t) \\ \vdots \\ \hat{\tau}_{N-1}(t) \end{pmatrix}$$

and

$$\mathbf{T}(\hat{\mathbf{x}}(t)) = \begin{pmatrix} T_0(\hat{\mathbf{x}}(t)) \\ T_1(\hat{\mathbf{x}}(t)) \\ \vdots \\ T_{N-1}(\hat{\mathbf{x}}(t)) \end{pmatrix} \qquad \mathbf{C}(t) = \begin{pmatrix} \mathbf{c}_0(t) \\ \mathbf{c}_1(t) \\ \vdots \\ \mathbf{c}_{N-1}(t) \end{pmatrix} \tag{2.15}$$

then we can express (2.14) in matrix form as

$$\bar{\boldsymbol{\tau}}(t) = \hat{\boldsymbol{\tau}}(t) - \mathbf{T}(\hat{\mathbf{x}}(t-1)) + \mathbf{C}(t)\hat{\mathbf{x}}(t-1) \tag{2.16}$$

If we now define

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_0^2 & & & \\ & \sigma_1^2 & & \\ & & \ddots & \\ & & & \sigma_{N-1}^2 \end{pmatrix} \tag{2.17}$$

then we can also express the error function from (2.13) in matrix form:

$$\boldsymbol{\epsilon}(\mathbf{x}; t) = [\bar{\boldsymbol{\tau}}(t) - \mathbf{C}(t)\mathbf{x}]^T \boldsymbol{\Sigma}^{-1} [\bar{\boldsymbol{\tau}}(t) - \mathbf{C}(t)\mathbf{x}] \tag{2.18}$$

Chapter 3

Video Features

Now that we have seen what audio features we can use for source localization let us see how we can use visual features for this task. In the acoustic case the measurement comes from the GCC, in the visual case a face detector is used to get the position of the speakers face in the camera's image plane. This measurement is than again compared to the speakers estimated position projected onto the image plane of the camera, which results in a two-dimensional innovation vector.

3.1 Projection

For doing this projection we assume a simple pin-hole camera-model. Let $\mathbf{x}$ denote the 3-dimensional speaker position that is going to be projected onto the image plane $\mathbf{I}$ of the camera at position $\mathbf{t}$ with focal length f, as shown in Figure 3.1. This results in the 2-dimensional image point $\hat{\mathbf{x}}$. The camera's translation $\mathbf{t}$ with respect to the global 3D coordinate origin and rotation given by $\mathbf{R}$ define the extrinsic parameters of the camera. The intrinsic parameters given by the focal length f, the sensor pixel size p_x and p_y, and the principal point $\begin{bmatrix} c_x & c_y & 1 \end{bmatrix}^T$ make up the camera matrix $\mathbf{P}$ as follows [Pol00]:

$$\mathbf{P} = \begin{pmatrix} \frac{f}{p_x} & 0 & c_x \\ 0 & \frac{f}{p_y} & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{3.1}$$

The camera's extrinsic and intrinsic parameters are determined by a calibration procedure like that of Zhang [Zha00]. Having all of the parameters together we can now express the projection of the position estimate $\mathbf{x}$ onto the image plane with the following equation:

$$\bar{\mathbf{x}} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix} = \mathbf{P}\mathbf{R}^T(\mathbf{x} - \mathbf{t}) \tag{3.2}$$

The corresponding 2-dimensional point is given by

$$f(\mathbf{x}) = \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} \frac{\bar{x}_1}{\bar{x}_3} \\ \frac{\bar{x}_2}{\bar{x}_3} \end{pmatrix} \tag{3.3}$$

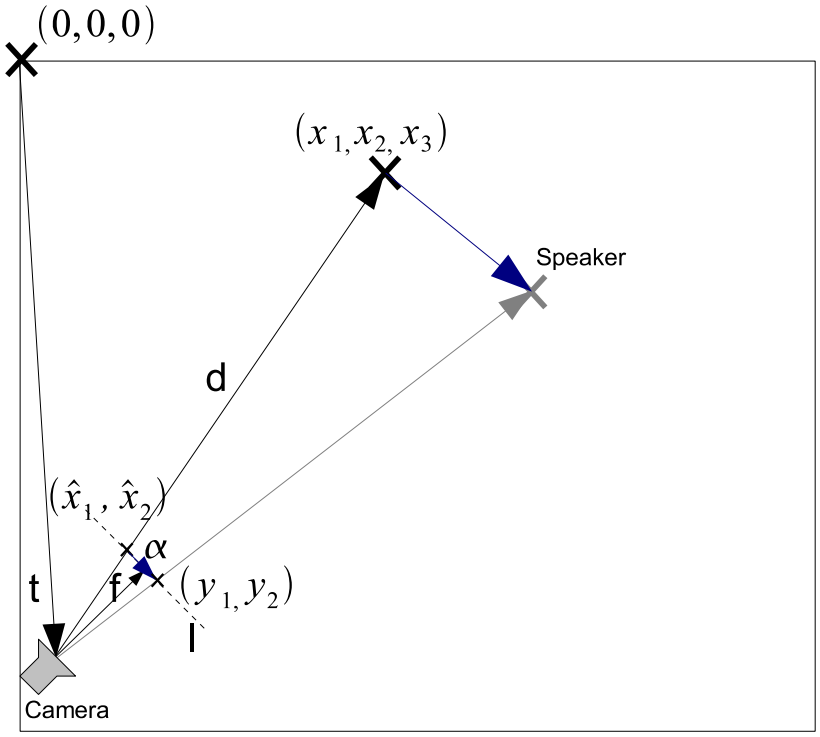

Figure 3.1: Projection of the speaker's position onto the image plane of a camera.

Since the the term $\mathbf{PR}^T$ does not change over time we can precalculate this term for better efficiency and substitute by

$$\mathbf{A} = \mathbf{PR}^T \tag{3.4}$$

To estimate the speaker's position we need to minimize the error between the estimated position projected into the image plane and the detected position produced by the face detector. This leads to a minimizing step that is like that for the audio features in (2.7). Therefore, we have to derive a linearization for the projection function in (3.3). As for the audio features in (2.8) we will approximate $f(\mathbf{x})$ with a first order Taylor series expansion. Hence, we take the partial derivative of $f(\mathbf{x})$ with respect to $\mathbf{x}$

$$\mathbf{C} = \nabla_{\mathbf{x}} f(\mathbf{x}) \tag{3.5}$$

where

$$c_{ij} = \frac{a_{ij} - a_{3j}\hat{x}_i}{\bar{x}_3} \tag{3.6}$$

for $1 \leq i \leq 2$, $1 \leq j \leq 3$ and $\{a_{ij}\}$ are the elements of $\mathbf{A}$.

In Figure 3.2 we can see the the nonlinear projection function compared to its linear counterpart. The reference point used for the linearization in the figure is the center of the room $\begin{bmatrix}2950 & 3550 & 1700\end{bmatrix}$. The speaker is moving in the x-direction through the center. As we can see the error within 1 m about the reference point is modest. The RMS error for the two figures is 2.053 pixel in x and 0.962 pixel in y. The worst-case deviation is 1.54 % in x and 1.94 % in y. This leads once again to the conclusion that assuming an accuracy of 1 m to localize the speaker the linearization performs almost as good as the nonlinear projection function.

3.2 Face Detection

Knowing the projected estimated speaker position in the camera's image plane, we have to find a face in a region around that point. Thus we need to utilize a face detector that gives us the information where faces can be observed.

3.2.1 Face Detector

To detect a face in the camera image the face detector implemented in the opencv library [BKP05] is used. This is based on a statistical approach for object detection that uses simple Haar-like features and a cascade of boosted tree classifiers as statistical model [LM02, VJ01].

For training a training set is used that consists of "positive" and "negative" samples, meaning faces and non-faces. These images are normalized to a fixed size of 24x24. For detection a search window is slided through the image and it is checked whether an this area is similar to a face or not. Scaling the classifier allows the detection of faces of different sizes.

The features used for classification are described by a template, the position within the search window and the scale factor. In Figure 3.3 the extended set of 14 templates is shown. These templates are a combination of two or three black and white rectangles that are rotated by 45° if needed. The feature's value is calculated by a weighted sum over the pixel sum of the black and the whole area, where the weights are inversely

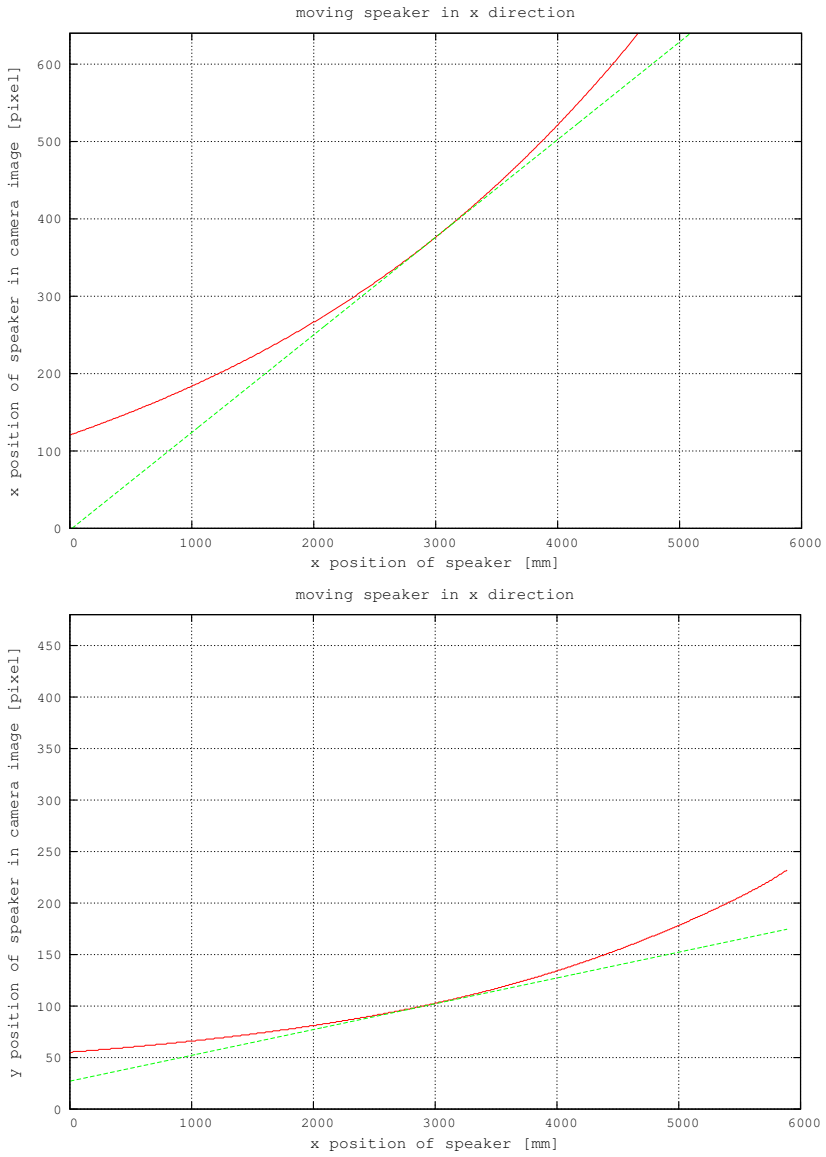

Figure 3.2: Comparison between nonlinear function and first order Taylor series. On the top the global x position of the speaker is plotted against the x position in the camera image. On the bottom it is plotted against the y position in the camera image.

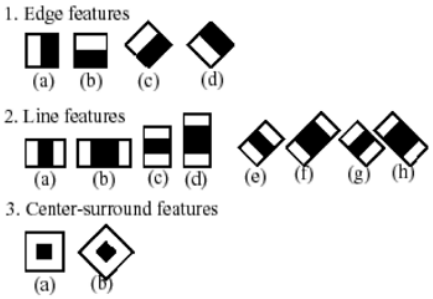

Figure 3.3: Extended set of Haar-like features

proportional to the corresponding area and of opposite signs. This results in following relation for example for feature 3(a) in Figure 3.3:

$$w_{\text{black}} = -9 \cdot w_{\text{whole}} \tag{3.7}$$

Since calculating pixel sums over multiple small rectangles for hundreds of features makes detection very slow, an *intergral image* [VJ01] is first calculated for the whole image I. This is a *Summed Area Table* (SAT), where each pixel represents the pixel sum of the rectangle from $(0, 0)$ to the pixels position (X, Y):

$$\text{SAT}(X, Y) = \sum_{x<X, y<Y} I(x, y) \tag{3.8}$$

thus allowing a arbitrary rectangle $r = \{(x, y)|x_0 \leq x < x_0 + w, y_0 \leq y < y_0 + h\}$ to be calculated as:

$$\begin{aligned} \text{RecSum}(r) \quad = \quad & \text{SAT}(x_0 + w, y_0 + h) - \text{SAT}(x_0 + w, y_0) - \\ & \text{SAT}(x_0, y_0 + h) + \text{SAT}(x_0, y_0) \end{aligned} \tag{3.9}$$

For the templates that are rotated by 45° a seperate integral image is calculated in the same way.

Now that the pixel sums can be efficiently calculated, the feature value can be obtained with the following equation:

$$x_i = w_{i,0}\text{RecSum}(r_{i,0}) + w_{i,1}\text{RecSum}(r_{i,1}) \tag{3.10}$$

where $w_{i,0}$ and $w_{i,1}$ are the weights for the pixel sums over the black and the whole area in the templates respectively.

The feature value is then put into a simple decision tree classifier with two teminal nodes:

$$f_i = \begin{cases} +1, & x_i \geq t_i \\ -1, & x_i < t_i \end{cases} \tag{3.11}$$

or three terminal nodes:

$$f_i = \begin{cases} +1, & t_{i,0} \leq x_i < t_{i,1} \\ -1, & else \end{cases} \tag{3.12}$$

where $+1$ corresponds to a face and -1 to a non-face.

Such a classifier doesn't detect a face, but rather a simple feature of it like eyes or nose. Therefore it is called a *weak classifier*. To detect a face many weak classifiers are put together to a complex and robust classifier by a procedure called *boosting* [FS96].

A boosted classifier is built by iteratively summing up weak classifiers and adjusting the weights for each one according to the error the classifier gives on the training set:

$$F = sign(c_1 f_1 + c_2 f_2 + \ldots + c_n f_n) \tag{3.13}$$

The weight c_i for a newly added weak classifier is assigned a higher value the smaller the error is. The weights of the other weak classifiers is then updated in such a way that the samples misclassified by the current boosted classifier are emphasized. This is proven [FS96] to achieve arbitrarily high hit rates and arbitrarily small false alarm rates if f_i is more selective than a trivial classifier that selects a feature with a probability of 50%.

But to achieve that performance a large testing set and many weak classifiers would be needed that would result in slow processing speed; Viola [VJ01] proposed cascading several boosted classifiers F_k that increase in complexity with greater k so that during detection the classification can stop at an earlier stage if the classifier failed to the detect the appropriate features there. A face is only detected if all F_k classify the image as a face. In experiments [BKP05] between 70% and 80% of the candidates are rejected in the first two stages having about 10 weak classifiers each. Thus this results in a great speedup of the detection, since not all classifiers are run over all candidates, but only on real faces.

3.2.2 Adaptive Background Model

To improve the detection rate a simple background model is used. It computes for each pixel the median over the last n camera images. The result is a background image that should ideally consist only of things that are not of interest.

Since the face detector is running only on a part of the whole image the median is only calculated for that area to save processing time.

If the absolute of the difference between a pixel value of the current image I and its corresponding value in the background image B is lower than a given threshold t it is classified as *background pixel*, otherwise as *foreground pixel*.

$$f(I,i,j) = \begin{cases} 0, & |I(i,j) - B(i,j)| < t \\ 1, & else \end{cases} \tag{3.14}$$

Therefore the probability of the detected face to be a valid face can be denoted as the ratio between the foreground pixels and the total number of pixels in the detected area:

$$p = \frac{\sum_{x \le i < x+w_f} \sum_{y \le j < y+h_f} f(I,i,j)}{w_f \cdot h_f} \tag{3.15}$$

where (x, y) is the position of the upper left corner and (w_f, h_f) the size of the detected face.

Thus a threshold can be set on the probability p to filter out background objects from the detected faces.

In Figure 3.4 the original image is shown beside the background model and the foreground segmentation, where the background model is updated every second. It

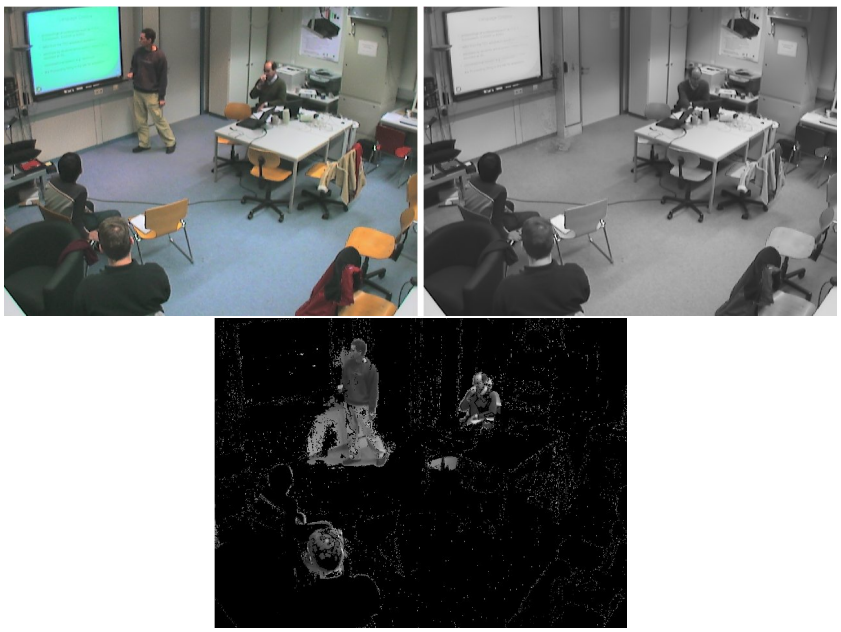

Figure 3.4: On the top left is the original camera image, on its right side the adapted background image that is updated every second and on the bottom the resulting foreground image (black being the background).

shows that it gives good results when the speaker is moving, but one disadvantage is that if the speaker stands still for a while he gets adapted into the background, just as the audience does when it is not moving.

Chapter 4

Speaker Tracking

Now that we have all features together that are used for speaker localization we need to combine them to get the 3-dimensional speaker position. For that a Kalman Filter is used that is then incrementally updated with the observations of the different sensors [Wel96, WB97]. Since the standard Kalman Filter is not able to process nonlinear measurement functionals the Extended Kalman Filter (EKF) [Hay02, §10] is chosen. To minimize the error introduced by the linearization, the EKF is further improved by introducing local iterations in the estimation step which leads to the Iterated Extended Kalman Filter (IEKF).

4.1 Theory of Kalman Filters

The Kalman Filter is described by a process and observation equation:

$$\mathbf{x}(t+1) = \mathbf{F}(t+1,t)\,\mathbf{x}(t) + \boldsymbol{\nu}_1(t) \tag{4.1}$$

$$\mathbf{y}(t) = \mathbf{C}(t,\mathbf{x}(t)) + \boldsymbol{\nu}_2(t) \tag{4.2}$$

The process equation (4.1) models the evolution of the state $\mathbf{x}(t)$ of the Kalman Filter from time t to $t+1$ by using the *transition matrix* $\mathbf{F}(t+1,t)$ and the process noise $\boldsymbol{\nu}_1(t)$. The observation equation (4.2) on the other hand specifies how the state $\mathbf{x}(t)$ is transformed into the observation space by the functional $\mathbf{C}(t,\mathbf{x}(t))$ under the presence of the observation noise $\boldsymbol{\nu}_2(t)$ as the measurement $\mathbf{y}(t)$. The process and observation noise are assumed to be zero mean with covariance matrices $\mathbf{Q}_1(t)$ and $\mathbf{Q}_2(t)$.

We now want to estimate the state that leads to the observation based on all the observations $\mathcal{Y}_{t-1} = \{\mathbf{y}(n)\}_{n=0}^{t-1}$ up to time $t-1$. Therefore we first predict state estimate $\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})$ and calculate the *innovation* $\boldsymbol{\alpha}(t,\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))$ at time t defined as:

$$\boldsymbol{\alpha}(t,\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})) = \mathbf{y}(t) - \mathbf{C}(t,\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})) \tag{4.3}$$

where $\mathbf{y}(t)$ is the observation and $\mathbf{C}(t,\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))$ is the predicted observation, given as the projection of the predicted state estimate into the observation space. In the Kalman Filter the *Kalman gain* weights this innovation to smooth out the measurement noise. The Kalman gain is defined as:

$$\mathbf{G}_f(t,\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})) = \mathbf{K}(t,t-1)\mathbf{C}^T(\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))\mathbf{D}(t,\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))^{-1} \tag{4.4}$$

where $\mathbf{C}(\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))$ is the linearization of the nonlinear functional $\mathbf{C}(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))$ about the current state estimate $\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})$ and $\mathbf{D}(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))$ is the *innovation covariance matrix* that is given by:

$$\mathbf{D}(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})) = \mathbf{C}(\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))\mathbf{K}(t, t-1)\mathbf{C}^T(\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})) + \mathbf{Q}_2(t) \tag{4.5}$$

If we now combine the previous state estimate $\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})$, the innovation $\boldsymbol{\alpha}(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))$ and the *Kalman gain* $\mathbf{G}_f(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))$, we get the new state estimate $\hat{\mathbf{x}}(t|\mathcal{Y}_t)$:

$$\hat{\mathbf{x}}(t|\mathcal{Y}_t) = \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}) + \mathbf{G}_f(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))\boldsymbol{\alpha}(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})) \tag{4.6}$$

Both the Kalman gain and the innovation covariance matrix depend on the *predicted state error covariance matrix* $\mathbf{K}(t, t-1)$ that is recursively updated together with the *filtered state error covariance matrix* $\mathbf{K}(t)$ by the following *Riccati equations*:

$$\mathbf{K}(t) = [\mathbf{I} - \mathbf{G}_f(t)\mathbf{C}(\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))]\mathbf{K}(t, t-1) \tag{4.7}$$

$$\mathbf{K}(t+1, t) = \mathbf{F}(t+1, t)\mathbf{K}(t)\mathbf{F}^T(t+1, t) + \mathbf{Q}_1(t) \tag{4.8}$$

This state error covariance matrix tells us how accurate the current state estimate is.

In the case of the IEKF [Jaz70, §8.3], we now replace (4.3) and (4.6) with local iterations, in which the error that is introduced by the linearization of the nonlinear measurement functional is reduced by iteratively approximating the position estimate to have a more accurate reference point for the linearization. Therefore, the position estimate $\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})$ is subsituted in (4.3) by a local position estimate $\boldsymbol{\eta}_i$ for the i-th iteration, which leads to following equations:

$$\boldsymbol{\alpha}(t, \boldsymbol{\eta}_i) = \mathbf{y}(t) - \mathbf{C}(t, \boldsymbol{\eta}_i) \tag{4.9}$$

$$\boldsymbol{\zeta}(t, \boldsymbol{\eta}_i) = \boldsymbol{\alpha}(t, \boldsymbol{\eta}_i) - \mathbf{C}(\boldsymbol{\eta}_i)[\hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}) - \boldsymbol{\eta}_i] \tag{4.10}$$

$$\boldsymbol{\eta}_{i+1} = \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}) + \mathbf{G}_f(t, \boldsymbol{\eta}_i)\boldsymbol{\zeta}(t, \boldsymbol{\eta}_i) \tag{4.11}$$

This local iteration is initialized by setting

$$\boldsymbol{\eta}_1 = \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}) = \mathbf{F}(t, \hat{\mathbf{x}}(t-1|\mathcal{Y}_{t-1})) \tag{4.12}$$

The local iteration is stopped as soon as the difference between $\boldsymbol{\eta}_i$ and $\boldsymbol{\eta}_{i+1}$ drops below some threshold. In case the local iteration is run only once, the IEKF reduces to an extended Kalman Filter, since $\boldsymbol{\eta}_2 = \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})$ as defined in (4.6). In each local iteration both $\mathbf{G}_f(t, \boldsymbol{\eta}_i)$ and $\mathbf{C}(\boldsymbol{\eta}_i)$ are updated and after the last iteration $\hat{\mathbf{x}}(t|\mathcal{Y}_t)$ is set to $\boldsymbol{\eta}_f$. As Jazwinski [Jaz70, §8.3] reports this way provides faster convergence when the measurement functional is significantly nonlinear or the initial state is very inaccurate.

Figure 4.1 shows that with increasing inaccuracy more iterations are used. These inaccuracies can come from a fast moving speaker or a long time between two updates.

4.2 Refinements

Here we consider several refinements of the speaker tracking to improve speed and accuracy of the estimation.

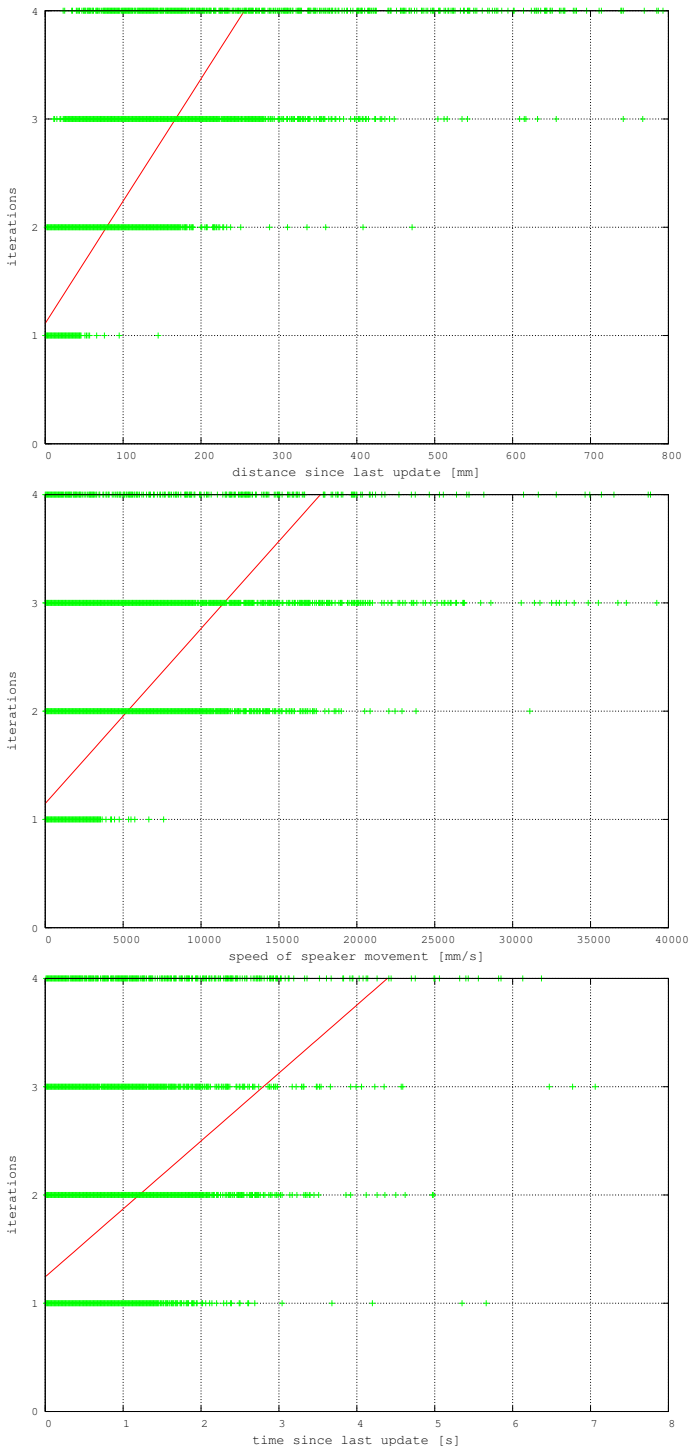

Figure 4.1: Relationship between the number of iterations used and the distance between two updates, speed of the moving speaker and time since last update.

4.2.1 Innovation Filter

It often happens that for example the audio feature focuses on some background noise like some keyboard clicking or so then it is useful to have a mechanism that ignores such outliers. This can be easily achieved by checking if the innovation is inside the ellipsoid given by the innovation covariance matrix as shown in Figure 4.2. For that the

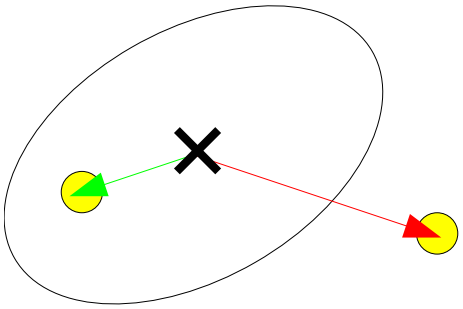

Figure 4.2: This figure shows 2 observations (yellow circles) from which one innovation is ignored (red arrow) and one is valid (green arrow), because it falls within the validation region of the innovation covariance.

Mahanalobis distance of the innovation (4.3) using the innovation covariance matrix given in (4.5) is calculated and compared to the gating threshold γ:

$$d^2 = \boldsymbol{\alpha}(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))^T \mathbf{D}(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))^{-1} \boldsymbol{\alpha}(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1})) \leq \gamma \tag{4.13}$$

d is also called *normalized innovation*. Thus an estimate can be simply ignored if the the squared normalized innovation d^2 is lower or equal to the gating threshold γ. The gating threshold depends on how strict the filter should be. It is a scale factor for the size of the gating ellipsoid.

4.2.2 Dynamic Search Window

To increase the processing speed of the Face detector (3.2.1) the area of interest on which the face detector is run is restricted by a search window. This window is dynamically calculated as bounding box of the innovation covariance matrix (4.5) scaled by some factor and increased by the expected face bounding box, as shown in Figure 4.3. The extent of the bounding box of the innovation covariance is calculated as the double of the square-root of the diagonal elements of $\mathbf{D}(t, \hat{\mathbf{x}}(t|\mathcal{Y}_{t-1}))$. Let us denote the width w and height h of the search window as follows:

$$w = 2s\sqrt{D_{11}} + f_x \tag{4.14}$$

$$h = 2s\sqrt{D_{22}} + f_y \tag{4.15}$$

where (f_x, f_y) is the face image size that is expected to be detected and s the factor used to scale the innovation covariance. The expected face size is estimated as:

$$f_x = \frac{a \cdot f}{p_x \cdot d} \tag{4.16}$$

$$f_y = \frac{a \cdot f}{p_y \cdot d} \tag{4.17}$$

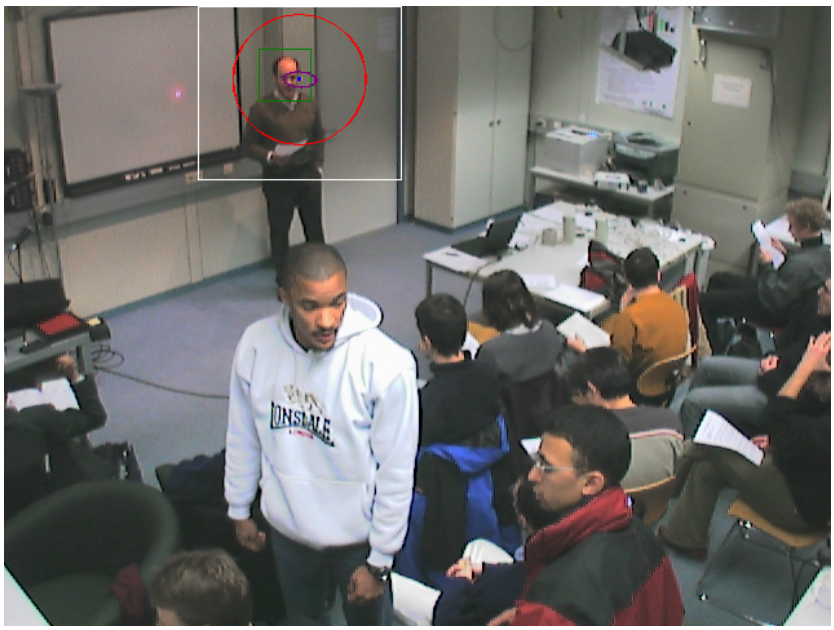

Figure 4.3: The white box is the search window used for face detection, where the green box is one detected face. The red ellipse is the innovation covariance and the purple ellipse the predicted state error covariance projected into the camera space.

where f is the focal length, (p_x, p_y) the pixel size of the camera, a is the expected extend of a face in mm and d the distance between the camera and the current estimated speaker position.

Chapter 5

Experiments

To see how good the proposed approach works, it was evaluated on approximately three hours of audio and video data recorded during seven seminars by students and faculty at the University of Karlsruhe (UKA).

The recording setup consisted of four T-shaped microphone arrays with four elements each and four video cameras in the corners of the room (5.1). The four cameras of the room were calibrated with the technique of Zhang [Zha00].

For the purpose of evaluation the centroid of the speaker's head was manually marked every 0.7 second in the images from four video cameras. These 2 dimensional labels have then been combined to 3 dimensional speaker positions using triangulation as described in [FS02] resulting in the "ground truths" that are accurate to within 10 cm.

Since the seminars were recorded in an environment that is used both by the seminar participants and students as well as staff engaged in other activities, these recordings are realistic test sets for acoustic as well as visual source localization. Because of the nature of farfield sensors the recordings contain besides speech also noise from fans, computer, doors and cross-talk from other people present in the room.

Table 5.1 shows the results of a comparison between audio-only, video-only and audio-video experiments.

We can see that video-only the estimation accuracy is very poor. This is a result of the fact that it often occurs that in different camera views different faces are focused, because the desired face is not visible or not detected in the other view and another person's face falls into the current search window, that is then tracked from then on. So the estimated position converges to a point in the room where either no person or at least not the desired person is located. As soon as this happens the speaker tracking algorithm loses track of the desired speaker and hardly returns to the correct position. Another point is that face detection alone is a very simple video feature, that does

	RMS Error (cm)				
Tracking Mode	X	Y	Z	2D dist	3D dist
audio-only	34.9	40.7	12.9	55.5	57.2
video-only	40.9	54.8	13.0	71.4	72.8
audio-video	30.9	36.9	8.1	49.0	49.9

Table 5.1: Root mean square (RMS) errors for source localization algorithms.

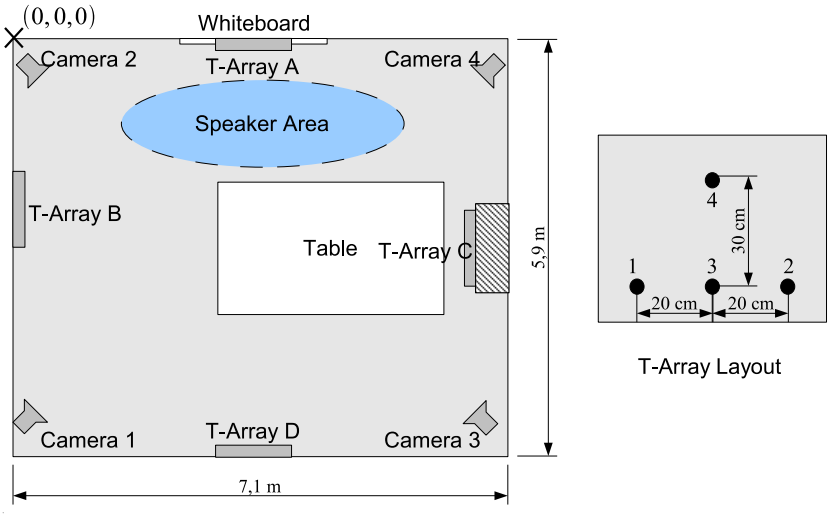

Figure 5.1: Sensor setup: four video cameras in the room corners and four T-Arrays with four elements each as shown on the right.

not necessarily lead to a good tracking system. But as we can see, combined with the audio, it is enough to get an improvement over the audio-only tracking system. Since face detection is done in the image plane, a detection of the face in only one camera view over a long time, due to misdetections or occlusion in the over views, is not enough to estimate a 3D position correctly, and leads to divergence and makes the system unobservable [Wel96, WB97].

The audio-only mode has better accuracy than video-only, because in this task there is rarely any other person than the desired one speaking and acoustic noise sources in the room result mostly in small inaccuracies of the estimation, but not to lost tracks. But there is still the problem that, when the speaker is not speaking for a while the state error covariance grows larger and after some point we don't really have a clue anymore about where the speaker really is. So we cannot rely on the estimates when there is no speech present.

The approach that combines both audio and video features results in a greater accuracy than the audio only approach, since it is capable of holding track of the desired speaker even if he is not speaking, when his face is still detected, and also if he is not visible or detected in all camera views, while he is speaking. Thus it rarely loses track of the speaker, even though it can still happen in cases where there is no speech and the system falsely tracks different faces in different views, thus reducing to a video-only system at that moment.

The parameters chosen for the audio-video experiment were also used to run the experiments on a single modality. The system was initialized with a fixed starting position for all seminars so the Kalman filter had to converge to the true position. The IEKF was iterated at most 5 times. The process noise was set to (154.62, 184.13, 34.24). All position estimates that were outside the room have been ignored. Furthermore a innovation filter was used with a gating threshold of 4.0 to get rid of outliers.

On the audio side all time delays that were outside the valid range given by the geometry of the room were ignored. Further a threshold of 0.18 was set on the maximum peak of the GCC for each microphone pair, so that only those pairs were used

in the estimation process that exceeded that threshold. Only two of the four T-Arrays were used, namely T-Array B and D, and of these all possible microphone pairs where processed with a measurement noise of 0.2 ms.

On the video side all four cameras have been used with a measurement noise of 25 pixels. The scale factor for the face detector's dynamic search window was set to 2.0 and the expected extend of a face to 50 cm.

Chapter 6

Conclusions and Future Work

As we saw in Table 5.1 source localization using both audio and video features works better than using only a single set of features. The bad video only results are coming from loosing track of the speakers face after some time from which the algorithm cannot recover.

Also note that the proposed approach integrates position estimation into the Kalman filter rather then using closed-form position estimations.

Through the architecture of sequential updating with different sensors real-time capability can be easily achieved by simply using a well chosen scheduling mechanism to choose from the pool of sensors. This makes it also possible to use more microphone pairs than would normally be possible when using closed-form estimations where all data has to be available at once.

Since the technique was not evaluated on artificial or simulated data, but rather on real seminar data, its useability in a real environment is more likely met.

For future research it makes sense to look for better and improved video features that perform well also when used alone. Then the overall performance would probably also be improved.

To improve the quality of the audio features improved speech activity detection and noise reduction techniques have to be more closely looked at.

Chapter 7

Acknowledgments

First of all I would like to thank my advisor John McDonough for his great guidings, support and ideas for this thesis and his help in understanding all the complicated stuff. I also thank Ulrich Klee for his work on audio source localization, on which this thesis is based. Kai Nickel greatly helped me with all the visual stuff. I would like to thank Hazim Ekenel for giving me assistance with everything related to face detection and training the face detector. Many thanks also go to Julius Ziegler who helped me in understanding the principles of the Kalman filter and also had great ideas that helped me on. And last but not least I would like to give thanks to anyone else who helped me in the process of this work. I greatly appreciate the work with all of you.

Bibliography

[BES06] K. Bernardin, H.K. Ekenel, and R. Stiefelhagen. Multimodal identity tracking in a smartroom. In *3rd IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI)*, 2006.

[BKP05] Gary Bradski, Adrian Kaehler, and Vadim Pisarevsky. Learning-based computer vision with intel's open source computer vision library. *Intel Technology Journal*, 9, May 2005.

[CH94] Y. T. Chan and K. C. Ho. A simple and efficient estimator for hyperbolic location. *IEEE Trans. Signal Proc.*, 42(8):1905–15, August 1994.

[DZD01] Ramani Duraiswami, Dmitry Zotkin, and Larry Davis. Multimodal 3-d tracking and event detection via the particle filter. In *Workshop on Event Detection in Video, International Conference on Computer Vision*, pages 20–27, 2001.

[FS96] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[FS02] D. Focken and R. Stiefelhagen. Towards vision-based 3-D people tracking in a smart room. In *IEEE Int. Conf. Multimodal Interfaces*, October 2002.

[Hay02] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, New York, fourth edition, 2002.

[Jaz70] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

[JV03] M. Jones and P. Viola. Fast multi-view face detection. In *IEEE Conference on Computer Vison and Pattern Recognition*, 2003.

[Kay93] S. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[KC76] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust. Speech Signal Proc.*, ASSP-24(4):320–327, August 1976.

[KGMon] Ulrich Klee, Tobias Gehrig, and John McDonough. Kalman filters for time delay of arrival-based source localization. *EURASIP Special Issue on Multichannel Speech Processing*, submitted for publication.

[LM02] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP*, 2002.

[OS94] M. Omologo and P. Svaizer. Acoustic event localization using a crosspower-spectrum phase based technique. In *Proc. ICASSP*, volume II, pages 273–6, 1994.

[Pol00] Marc Pollefeys. *Tutorial on 3D Modeling from Images*. Katholieke Universiteit Leuven, 2000.

[SA87] J. O. Smith and J. S. Abel. Closed-form least-squares source location estimation from range-difference measurements. *IEEE Trans. Acoust. Speech Signal Proc.*, ASSP-35(12):1661–9, December 1987.

[SSR01] N. Strobel, S. Spors, and R. Rabenstein. *Joint Audio-Video Signal Processing for Object Localization and Tracking*. Springer Verlag, Heidelberg, Germany, 2001.

[Van02] H. L. Van Trees. *Optimum Array Processing*. Wiley-Interscience, New York, 2002.

[VJ01] Paul Viola and Michel J. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vison and Pattern Recognition*, 2001.

[WB97] G. Welch and G. Bishop. SCAAT: Incremental tracking with incomplete information. In *Proc. Computer Graphics and Interactive Techniques*, August 1997.

[Wel96] Gary Francis Welch. *SCAAT: Incremental Tracking with Incomplete Information*. PhD thesis, University of North Carolina, Chapel Hill, NC, 1996.

[WNM05] Matthias Wölfel, Kai Nickel, and John McDonough. Microphone array driven speech recognition: influence of localization on the word error rate. In *Proc. Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI05), Edinburgh*, 2005.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis Machine Intel.*, 22:1330–1334, 2000.