Delft University of Technology

Department of Multimedia and Signal Processing

December 2, 2013

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER SUPERVISION BY

Richard Heusdens & Nikolay Gaubitch

ENTITLED

Improving indoor localization of Android phones

For the use of speech enhancement by using a different calibration signal for each phone and including the relative clock skew

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

Thesis Advisor
Dr.ir. R. Heusdens

Thesis Reader
Prof.dr.ir. G.J.T. Leus

Chairman of Department
Prof.dr.ir. R.L. Lagendijk

Thesis Reader
Dr.ir. G.J.M. Janssen

Thesis Reader
Dr. N.D.Gaubitch

Improving indoor localization of Android phones

For the use of speech enhancement by using a different calibration signal for each phone and including the relative clock skew

By

Erik Roeling B.Sc.

Thesis

Submitted in Partial Fulfilment of the Requirements
for the Degree of Master of Science
in the School of Engineering at
Delft University of Technology, 2013

Delft, The Netherlands

Acknowledgments

Hereby, I would like to mention and thank a number of persons who directly or indirectly contributed to my thesis. They contributed to enriching the ideas present in the thesis and to the ways of expressing them. In the first place I would like to thank my supervisors Richard Heusdens and Nikolay Gaubitch for supporting me during the whole process of it. Starting from the theoretical work and ending in the writing part. Richard was always very friendly, guiding me when I lost the overview, answering my questions and never losing his patience. Nikolay was always a great help in problem solving from an extra USB-cable to a complete new look on a problem he supplied it all.

Next to the supervisors there were a lot of people that supported my throughout the whole time I have been at the university. They have given me opportunities for a live time. Unfortunately there is no space to mention them all. Two groups I would like to mention especially. In the first place: the audio group, on the 11th floor, and the students present in the student room during my stay there. I have enjoyed the conversations and the Friday evenings in the /Pub while having a drink[1] and a pizza. Secondly I would like to mention, the 139th board of the "Electrotechnische Vereeninging" this group of five people, myself not included, have supported me during all the troubles on the road to this point. Without them my stay at the university would have been totally different.

Off course I cannot forget all my friends for the interesting conversations online, in the studentroom, at the ETV and in the Pub. Last but not least, I would also like to thank my family, who always supported me throughout my stay at the university.

Erik Roeling BSc.
December 2, 2013

[1] or two

Improving indoor localization of Android phones

Erik Roeling B.Sc.

Department of Multimedia and Signal processing
Delft University of Technology
Delft, The Netherlands
2013

ABSTRACT

In this thesis an existing method to localise phones in an indoor environment is extended. For this a new method to synchronise the clocks in presented. When looking at the different causes of localisation errors we conclude that the incorrect detection of the time of arrival (TOA) gives the largest errors. This is why we focus on the detection of the correct TOA in this thesis. We discuss the effect of different kinds of calibration signals on the estimation of the TOA. It is found that a chirp signal can be used to estimate the TOA at a sufficient accuracy.

The errors caused by the distance-estimation method used, occur if the microphone and loudspeaker on a phone are not co-located. The error is only present when the different phones have different orientations. In the best case, the microphone positions can be estimated at accuracy less than five millimetres. A direct relation between the maximum error and the distance between the microphone and loudspeaker on the phones is found. In a computer simulation the error only exceeds an average of 5 centimetres when 10 or more phones are used, placed at random locations and orientations. These results are confirmed by real-data experiments.

Table of Contents

List of Figures

List of Symbols

i	Receiver i
j	Source j
$.^{(r)}$	Receiver
$.^{(s)}$	Source
ν	Noise (acoustical and quantization)

Constants

c	Speed of sound
f_S	Sampling frequency
$\mathcal{I}$	Number of receivers
$\mathcal{J}$	Number of sources

Location Variables

r_i	Receiver location vector $[x\ y\ z]^T$
s_j	Source location vector $[x\ y\ z]^T$
R	Receiver location matrix ($\mathcal{I}$ x 3)
S	Source location matrix ($\mathcal{J}$ x 3)

Time Variables

Δ_i	The clock skew
δ_i	The clock offset
ϵ^*_{ij}	Internal delays
τ_j	True time an event occurred
q	Noise due to sampling
t_{ij}	Time at which event was registered by device
T_{int}	Time between two events

Chapter 1

Introduction

To organise a teleconference, a teleconference system is needed. These systems either use temporal or spatial processing for enhancing the speech quality. For temporal processing, where only one microphone is available, the recorded audio is processed in short frames of a few milliseconds that are converted to the frequency domain. In the frequency domain a gain function is applied to the DFT coefficients. This function suppresses frequency bands of the recorded audio with a lower speech to noise ratio more than the bands with a higher speech to noise ratio. Because of the processing of individual frames spectral peaks may remain, which may be perceived as "musical noise" [1]. Temporal enhancement systems include: spectral subtraction and systems based on linear and non-linear minimum mean-square error estimators [2, 3].

For spatial processing, multiple microphones are needed. With this either adaptive noise cancellation or beamforming can be used to enhance the noisy speech. With adaptive noise cancellation one or more of the available microphones are used to record a noise reference. The reference signal is filtered with an adaptive filter in order to subtract it from the noisy speech. This method is an optimal way to filter the noisy speech as long a speech free reference is available [4]. When beamforming is used, the recordings of the different microphones are combined constructively for the desired speech signal and destructively for all the other signals. This way the target speech signal is enhanced while other sources are rejected. To be able to do beamforming, the relative Time of Arrivals (TOA)s of the speech signal needs to be available. For this to work the relative locations of the microphones are needed [5].

In this thesis we present an updated method to localise the microphones of several Android smart-phones, with the purpose to improve the quality of captured speech. Because just under 70% of all smart-phones run on Android (May 2013) [6] it is possible to set-up a teleconference system on the fly, if the method works on Android phones. This would mean that teleconferences are not fixed to certain locations, which reduces the price to organise a teleconference. Besides the large market share, Android is an open source platform which gives us the possibility to create our own applications.

As the title of this thesis suggests, the goal of the contribution is to preform localisation of randomly distributed microphones. In order to explain the proposed extension, the problem first needs to be formulated. This will be done in Section 1.1. In Section 1.2, the experimental set-up for both computer simulated and real-data measurements will be described. For the outline of this thesis see Section 1.3.

1.1 Problem formulation

To be able to determine the relative locations of the microphones we need to know the distance between them. We use audio to determine distances, because other signals would measure the distance between other sensors and we are only interested in the location of the microphones. The distance that can be measured is the distances between the loudspeakers and microphones. See Fig. 1.1 for an example for a two phone case. The distance between the microphone of the ith phone and the loudspeaker of the $j$the phone is denoted by d_{ij}. So $d_{ij} = ||r_i - s_j||$, where r_i is the microphone location ($[x\ y\ z]^T$), s_j is the loudspeaker location ($[x\ y\ z]^T$) and $||.||$ denotes the Euclidean norm. In the figure the distance between the microphone and loudspeaker on one phone is quite large; the actual distance depends on the brand of the phone. The distances range from 2 cm up to 12 cm. The number of microphones/devices is defined as $\mathcal{I}$ and the number of loudspeakers/sound sources as $\mathcal{J}$ so $i \in 1, 2, 3, \ldots, \mathcal{I}$ and $j \in 1, 2, 3, \ldots, \mathcal{J}$.

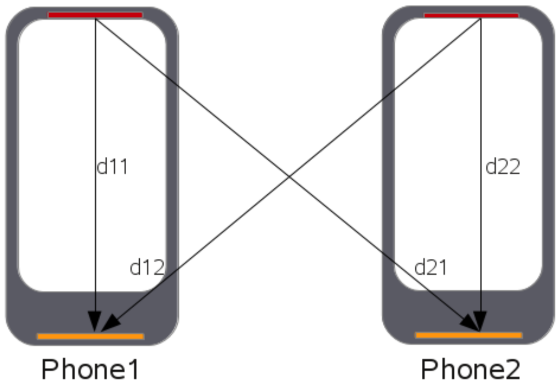


Fig. 1.1: Set-up with two phones with all possible distances inserted.

Once the distances between the microphones and loudspeakers are estimated, the relative locations of the microphones can be estimated. But for this at least four phones are needed (in 3D)[7].

1.1.1 Distance estimation

The distance is determined by the propagation time of an audio signal between a loudspeaker and a microphone. The sound source starts emitting a signal at $\tau^{(s)}$, this we call the Time of Departure (TOD). The signal arrives at $\tau^{(r)}$, this is the TOA. They are related to each other by the distance between the microphone and the loudspeaker and the speed of sound. If only the direct path is considered, we have

$$\tau_{ij}^{(r)} = \tau_j^{(s)} + \frac{d_{ij}}{c}, \tag{1.1}$$

where c is the speed of sound. In this thesis we use $.^{(r)}$ to denote received and $.^{(s)}$ to denote send.

In practice, there are multiple paths between the loudspeaker and the microphone, all paths together we call the channel. The multiple paths are caused by reflections. The actual recording is affected by the reverberation and acoustical noise following

$$x_{ij} = h_{ij} * s_j + \nu_{ij}, \tag{1.2}$$

where x_{ij} is the recorded signal, h_{ij} is the channel, s_j is the send signal and ν_{ij} is acoustical and quantization noise.

The channel, and thus amount of reflections that appear at the receiver, depends on the room. The signal can get reflected by the walls, ceiling, floor and any object in the room there by the intensity gets less since energy is dissipated. The signal is also attenuated while propagating trough the room.

The rate at which the intensity of the signal drops is defined by the time it takes for the intensity to drop with 60 dB with respect to the direct-path signal intensity. This time is called the *reverberation time* (RT_{60}). The reverberation time relates to the room following Sabine's equation [8]

$$RT_{60} = \frac{24 \ln 10}{c} \frac{V}{A\alpha}, \tag{1.3}$$

where A is the surface area of the room (m^2), V is the volume of the room (m^3) and α is the average absorption coefficient of the room surfaces. The equation does not take into account losses from the sound propagating through the air.

1.1.2 Estimation of TOD and TOA

The TOD and TOA, that are used to determine the distance between the loudspeakers and microphones, cannot be measured directly. Next to this, they are measured/estimated on different devices that may not use one "common" clock. This means that they cannot be compared to each other. We actually have three separate problems: the estimation of the TOD and TOA and the need for one "common" clock.

TOD estimation

At the sending side on a known moment $t^{(s)}$ the signal is to be sent. Then there is some unknown delay ($\epsilon^{(s)}$) that is caused by running processes and an internal delay path. This last one is mainly due to the conversion of the digital signal into an analogue signal. This two together give the actual TOD. If we put this in an equation we get:

$$\tau_j^{(s)} = t_j^{(s)} + \epsilon_j^{(s)}. \tag{1.4}$$

TOA estimation

At the receiving side the signal arrives at $\tau^{(r)}$. Then there is an internal delay ($\epsilon^{(r)}$) mainly due to the conversion of analogue signal to a digital signal. This conversion, that involves sampling, also results in some uniform distributed noise $q^{(r)}$ over the interval: $\left[-\frac{1}{2f_s}, \frac{1}{2f_s}\right)$, where f_s is the sampling frequency. After this the signal is stored at $t^{(r)}$. If this is put into an equation we get:

$$t_{ij}^{(r)} = \tau_{ij}^{(r)} + \epsilon_i^{(r)} + q_{ij}^{(r)}. \tag{1.5}$$

"Common" clock creation

The difference between two clocks can be described by two parameters. We define τ_w as an accurate real-time standard like UTC[1] and τ_p as the time on a phone. At some true global time t_0 there can be a difference between τ_p and τ_w. We define this difference as the offset (δ) according to

$$\delta = \tau_p(t_0) - \tau_w(t_0), \tag{1.6}$$

where $\tau_p(t_0)$ is the time on the phone at t_0 and $\tau_w(t_0)$ is the time according to the real-time standard at t_0

If the difference between τ_w and τ_p changes with time there also is clock skew Δ. The rate at which this difference changes is the clock skew. We define the skew as

$$\Delta = \lim_{t_1 \to t_0} \frac{\tau_p(t_1) - \tau_p(t_0)}{\tau_w(t_1) - \tau_w(t_0)}. \tag{1.7}$$

By combining the offset and skew in one equation we obtain the following relation between τ_w and τ_p,

$$\tau_p = \tau_w \Delta + \delta. \tag{1.8}$$

1.1.3 Distance estimation

By combining (1.8) with (1.4) and (1.5) we can use (1.1) to estimate the distance between the loudspeakers and microphones. The combining of (1.8) with (1.4) and (1.5) results in

$$\tau_j^{(s)} = \frac{t_j^{(s)} + \epsilon_j^{(s)} - \delta_j^{(s)}}{\Delta_j} \tag{1.9a}$$

$$\tau_{ij}^{(r)} = \frac{t_{ij}^{(r)} - \epsilon_i^{(r)} - q_{ij}^{(r)} - \delta_i^{(r)}}{\Delta_i}. \tag{1.9b}$$

[1]Coordinated Universal Time

By estimation the channel, the direct path can be found, i.e. the $t^{(r)}$ that belongs to the arrival of the signal that was sent, for this to work a line of sight is needed. The signals that are sent with as purpose to estimate the channel will be called calibration signals. How to estimate the channel and which signals give the best results will be discussed in Chapter 3. The estimation of δ, Δ and ϵ will be discussed in Chapter 4. This leaves q but this one cannot be estimated; averaging over multiple measurements can reduce the error caused by this value, a higher sampling rate or interpolation can also reduce the error.

1.1.4 Location retrieval

If all the distances between the microphones are found, classical multidimensional scaling (CMS) could be used to determine their relative locations. But as said the distance between the microphones and loudspeaker pairs is found. To find the relative locations of the microphones a singular value decomposition (SVD) [7] can be used. For this last method, a minimum of four microphones and four loudspeakers is needed in 3D, and one less of each in 2D.

1.2 Experimental environment

As mentioned before, we want to localise a set of microphones randomly distributed within a room. Since this application is to be used in a teleconference setting, we define the room to be a typical meeting room. With this in mind we assume the reverberation time to be between 0.15 and 0.7 seconds. The room size we used was 6 x 5 x 4 meters.

In this virtual room, a table of 1.5 m high was located in the centre. On that table, the phones for localisation where positioned in a random way. Each phone has a microphone and a loudspeaker. The distance between the microphone and loudspeaker on the phones is assumed to be known. We want to estimate the relative locations of the microphones. This is because these locations are needed to do beamforming. The error is defined as

$$e = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} ||\hat{r}_i - r_i||, \tag{1.10}$$

where $\hat{r}_i$ is the estimated location of microphone i. Since we are only interested in the relative locations the set of obtained locations can be rotated in any direction to minimize this error. We use Procrustes alignment to minimize this error.

1.2.1 Model

For the simulations a model was made in MATLAB® of the recording and sending part of Android phones together with a room impulse response to simulate the channel between the microphones and loudspeakers. For the model the speed of sound of 343 m/s was used and the same sampling rate as Android phones was used, 44.100 kHz. A distance of 3.1 centimetres between the microphone and loudspeaker on a phone is used. This is the distance for the Samsung phones.

For the room impulse response, the room impulse response generator[9, 10] was used. Sub cardioid microphones where used since this angular response is most likely, but we are not sure since the manufacturers do not supply the necessary information. The microphones are orientated to the ceiling this was also done in the live experiments.

For the on- and offset a random value is used, these values are limited to a maximum that can be chosen. The positions of the phones can be chosen, or a random configuration can be used. The number of phones is limited to the number of different calibration signals.

Furthermore one can select the length of the calibration signals, in this thesis times between 0.05 and 0.5 seconds where used, the time between two calibration signals was 0.05 up to 1 seconds. The reverberation time can also be chosen, times between 0.15 and 1 second where used in our experiments. The length of the impulse response was adjusted to the reverberation time, only the response for half of the length of the reverberation time was calculated. This means only a part of the complete room impulse response was obtained, since our interest is to check if no errors are made the only interesting part is the beginning with the relatively high pulses. The average pulse height drops exponentially so half way the average pulse height is 30 dB down, this is more than enough for our purpose. The last thing one can choose is the amount of adjective (acoustical) noise in the recordings, for this white noise was used. We can use white noise because the correlated part of the noise is already there by using the RIR generator. Noise is defined as everything but the direct path signal.

1.2.2 Software

For the live experiments, Android phones we used two Samsung phones (Galaxy SII) and one HTC (Sensation), with an application of our creation where used. The application is able to make a recording to a WAVE[2] file and playback WAVE files of calibration signals that have to be send. Once the recording is done the application sends the file to our server, from where it can be downloaded for further processing. The sever can also send commands to one or more phones. From MATLAB® one can send commands to the server. For the communication between the server and the phones the pushing service from Google, Google cloud Messaging for Android, is used. This means the Google API is used to send a message to the Google service. For an overview of the communication see Fig. 1.2. In appendix A a detailed description of the software can be found.

1.3 Outline

The outline of this thesis will be as follows:

- Existing methods for time synchronization, distance estimation and localisation are presented in Chapter 2.
- Methods to detect the time of arrival of a pulse are presented in Chapter 3.

[2]Waveform audio file format, .wav

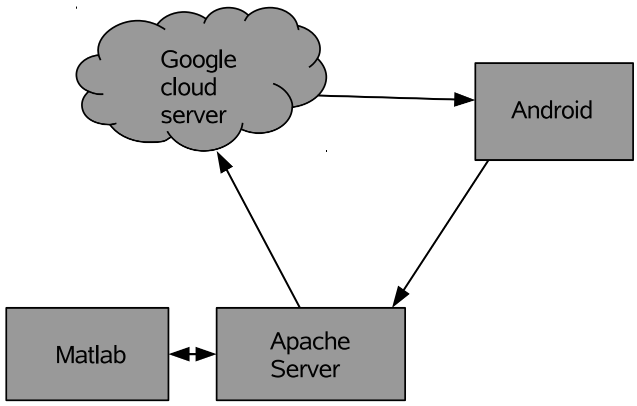


Fig. 1.2: The different blocks of the software with lines of communication between them.

- Methods to estimate of the different clock parameters of Android phones are presented in Chapter 4
- An existing method is improved with experiments the two methods are compared. The method and results are presented in Chapter 5
- Ideas for future work and the conclusions are presented in Chapter 6.

CHAPTER 2

Literature review

In this chapter, we review the existing literature on the topics relevant to the work presented in the remainder of this thesis. First the focus is on distributed time synchronisation. Secondly on distance estimation and finally we review the literature on localisation. There will only be a brief overview of the main contributions in the field that could be used to solve, a part of, our problem. A few methods extra relevant to the rest of the thesis will be discussed more in-depth.

2.1 Distributed time synchronisation

Plenty of work has been done on time synchronisation. Sundararaman [11] preformed a survey of the work done up to 2005, but this only looks at *physical clock synchronisation.* When using distributed systems, every device has its own physical clock and if time has importance for the task at hand, these clocks need to be synchronised. There are two types of synchronisation [11]. With the first type the device is synchronised with an accurate real-time standard like UTC. This type of synchronisation is called *physical clock synchronisation.* With the second type of synchronisation the clocks will be synchronised with respect to each other. This is called *logical clock synchronisation.* Both topics are of interest for this thesis and thus, will be explained.

For the our purposes we would like a ”common” clock that runs at an universal clock speed (physical), but there may be some offset compared to an universal clock (logical). This is why both topics are interesting. For skew estimation physical synchronisation is needed for offset estimation logical synchronisation is enough.

In [12] a method from Elson et al.called *Reference Broadcast Synchronisation* (RBS) is used to estimate the skew and offset. For this a reference packet was sent periodically over a network, while the TOA is estimated. When enough TOAs are received [12] tries to find a linear relation between them to estimate the skew. To estimate the true skew a device that really runs on the UTC clock has to send the pulses at some known interval. Since the true interval is known, any deviation is due to the clock skew. In practice this is accomplished by synchronisation in multiple layers. The first layer has access to a real time server, all lower layers synchronise with the layer above them. This is hard to accomplish on the Android phones since it is unlikely that one of the phones belongs to one of the layers, i.e. that one phone can, indirectly, synchronise with a real time server. The method can be used locally to synchronise the clocks of phones, but one ends up with the skew of the sending device.

For the estimation of the offset [11] discusses nine methods. This thesis will again only look at RBS because this is the only one that can also be used with audio signals. In general there is some server that knows the true time. The problem is to adjust/verify the clock of some other device over some network that causes an unknown delay. The delay path normally consists of some internal delays, the transfer to the other device(s) and finally post processing also resulting in an internal delay. As we have seen in Chapter 1 this is also the case with the audio signals we use. The RBS method for offset estimation decreases this delay path by sending to more than one device at the same time. Because RBS looks to the time difference of arrival the pre-processing time is not included in the error. As a result Elson was able to improve the estimation from $11\mu s$ to $1.85\mu s$. But to be able to do this, there is a need to have low level access to the network, since our goal is to implement this on an Android phone this is not a guaranteed. The principle of looking at the difference in TOA at the receiving side can be used if we implement this with calibration signals send via the loudspeakers.

2.2 Distance estimation

On distance estimation/localisation a good starting point is the survey by Liu et al.[13] on indoor localisation, and/or a survey by Pandey [14] for more general methods.

There are a serveral options for distance estimation. The most used are *Time of Flight* (TOF) and *Time of Arrival* (TOA), but others use the *Direction of Arrival* (DOA). A less commonly used method is based on the remaining energy in the received signal, *Received Signal Strength* (RSS). In the remainder of this section a brief description of the methods will be given along with a few references to actual implementations of the method.

2.2.1 Time of Flight

The idea behind TOF is very simple, but it requires special equipment. The idea is that a signal is sent out. This can be light, sound or even a network package. The TOF now is the time when this package returns to the source as a reflection. A high precision can be obtained in this way, and in the end you do not only know the distance to someone else, but actually the distances to obstacles / the walls of the room. In [15, 16] this is used for the localisation of a robot in a room. Although this can be used for localisation of the telephones it restricts the location of the microphones to the loudspeakers and this is not true in general. Furthermore the implementation of microphones and loudspeakers on typical phones are not ideal since the loudspeaker / microphone cannot rotate to face another direction. So is not possible to say from which direction to echo is coming.

2.2.2 Time of Arrival

For distance estimation by means of TOA, everything depends on the time at which a signal arrives at the destination. This is the most well know form of distance estimation. The difference between TOA and TOF is that for TOF the sender and receiver are the same device. this is not necessarilly the case for TOA estimation. There are a few different approaches within TOA estimation, one of those will be discussed separately since it is a major one, called Time Difference of Arrival (TDOA).

For TOA the distance is estimated from the difference between transmit time and the receive time. This means that there is need for synchronisation between the devices. TOA based distance estimation is used with various types of signals, in [17, 18] they use Wi-Fi while in [19, 20] a mobile network is used to find the minimum localisation error. These four implementations minimize the error in least squares (LS) sense

$$V(p) = (t - h(p))^T (t - h(p)), \tag{2.1}$$

where p it the position, h the channel, t the time of arrival and V the optimization criteria. This way a approximate estimation on the location is obtained, good enough for tracking someone but not for our purposes.

In [21] Chan uses a maximum likelihood (ML) approach to get a quadratic equation to solve from the different TOAs, but there is no simple solution. In [22, 23, 24] they use different methods, (Taylor's series, cancelling out terms and Fourier Transform, respectively) to make this quadratic problem linear, so it can be solved by means of LS. The obtained errors are in the order of meters. This is far from accurate enough since the distances between the microphones are in the same order of magnitude. In [25] a method that obtains an error in the order of centimetres can be found. In this method the squared distances are first estimated together with the clock parameters, after this a SVD based method is used to retrieve the locations of the microphones and sources. For this method to work there is a need of 4 phones for 2D and 5 phones for 3D. Since this method obtains an error in the order of centimetres, which is close to the best possible $7mm = c/f_s$, we will explain this method.

In [25] the error is minimized by a two-step approach. Where step one is the estimation of the squared distance and step two is the estimation of the clock parameters. These two steps are repeated over and over until the update rate is below some threshold. This decent can be ratter slow. The speed can be increased, by making the assumption that you perfectly know the relative times of the events, and thus know all TODs except τ_1. This can be done by sending pulses at a given interval.

The sending of the pulses for this method also gives a problem they have to be send from more than five different locations and thus an extra device is needed to send the pulses. This device itself is not localized and thus its location needs to be estimated after the location of all other devices is known. But when the relative onset times are known the method works with one phone less, so still four phones are needed for the localization (2D, and five for 3D).

The following assumption on the measured TOA is used

$$\frac{||r_i - s_j||}{c} = t_{ij}^{(r)} + \tau_j^{(s)} + \delta_i^{(r)}, \tag{2.2}$$

where $t_{ij}^{(r)}$ is the measured time of arrival the event j at the microphone of phone i, r_i is the receiver location vector $[x\ y\ z]^T$ and s_j is the source location vector $[x\ y\ z]^T$. $\tau_j^{(s)}$ is the TOD of event j and $\delta_i^{(r)}$ is the offset in the clock of node i. Note that we assume the same, but also include the clock skew. In this case we want to obtain r_i and s_j, for this we need to know δ. We chose τ and measure t_{ij}. From now on the speed of sound c is said to be one for simple calculations and $\tau_1^{(s)} = 0$ and $\delta_1^{(r)} = 0$. This can always be made true by knowing the distance between where event one occurred and the location of microphone one. In practice this means that one phone has to generate at least one event and that event is called event one.

By raising (2.2) to the power of two and subtracting $i = 1$ and $j = 1$

$$-2(\bar{r}_i)^T(\bar{s}_j) = T_{ij}^{(r)} + 2\tau_j^{(s)}(\hat{t}_{ij}^{(r)} - \hat{t}_{1j}^{(r)}) + 2\delta_i^{(r)}(\hat{t}_{ij}^{(r)} - \hat{t}_{i1}^{(r)}) + 2\delta_i^{(r)}\tau_j^{(s)} \tag{2.3}$$

is obtained, where

$$\begin{aligned} T_{ij}^{(r)} &= \hat{t^{(r)}}_{ij}^2 - \hat{t^{(r)}}_{1j}^2 - \hat{t^{(r)}}_{i1}^2 + \hat{t^{(r)}}_{11}^2 \\ \bar{r}_i &= r_i - r_1 \\ \bar{s}_j &= s_j - s_1 \\ i &= 2, 3, \ldots, \mathcal{I} \\ j &= 2, 3, \ldots, \mathcal{J}, \end{aligned}$$

Normally the factor $2\delta_i^{(r)}\tau_j^{(s)}$ gives a problem, since this forms a quadratic term, but if the events occurs at some interval, $\tau_j^{(s)}$ is known. To obtain a solution for (2.3), [25] uses the fact that the matrix $(\bar{r}_i)^T(\bar{s}_j)$ is of rank 3 (In 3D). If this is not the case this error must come from the unknown δ_i values. By minimizing the difference between the rank 3 approximation of $-2(\bar{r}_i)^T(\bar{s}_j)$ and $T_{ij}^{(r)} + 2\tau_j^{(s)}(\hat{t}_{a,ij}^{(r)} - \hat{t}_{1j}^{(r)}) + 2\delta_i^{(r)}(\hat{t}_{ij}^{(r)} - \hat{t}_{i1}^{(r)}) + 2\delta_i^{(r)}\tau_j^{(s)}$ a new estimation of $\delta_i^{(r)}$ is found. With this estimation a new value for $-2(\bar{r}_i)^T(\bar{s}_j)$ is calculated and the process is repeated. Once the difference between $(\bar{r}_i)^T(\bar{s}_j)$ and the rank 3 approximation of it is below a certain threshold the iteration is stopped.

With this method the offsets $\delta_i^{(r)}$ are estimated as well as a squared distance matrix $(\bar{r}_i)^T(\bar{s}_j)$.

Time Difference of Arrival

TDOA is used for tasks where there is no synchronisation possible between sender(s) and receiver(s), but there is synchronisation between the different sender(s) or receiver(s). For instance GPS[1] works this way. When using this method there is synchronisation between the senders. By using the location of the sources together with the TDOAs the location of the receiver can be estimated, for a detailed description please see [26].

In [27, 28] one will find techniques that obtain errors within 2 to 10 mm, but in our case there is no previous knowledge about the source locations. In [29, 30] one will find methods that do not include knowledge about this locations. There best case errors where quite different, in order: 6 cm and 1 cm. The localisation of laptops is done in [29], they use a ML estimation and Taylors expansion from [22] to obtain a result. The main disadvantage is that they need at least five devices. In [30] Hennecke localises Android phones. For this pairs of two phones are used with the rotation sensor of the phones as extra information. Since Hennecke also obtains a minimum error that is close to the best possible error his method will be explained more

[1] Global Positioning System

in-depth. Like [25], this is a complete method that performs both clock parameter estimation and distance estimation, before the estimation of the microphone locations.

Hennecke uses the known distance between the microphone and loudspeaker on Android phones and states that the location of the loudspeaker S can be expressed as

$$S = R + \begin{bmatrix} d_{11}\cos(\phi_1) & d_{22}\cos(\phi_2) & \dots & d_{\mathcal{I}\mathcal{I}}\cos(\phi_{\mathcal{I}}) \\ d_{11}\sin(\phi_1) & d_{22}\sin(\phi_2) & \dots & d_{\mathcal{I}\mathcal{I}}\sin(\phi_{\mathcal{I}}) \\ 0 & 0 & \dots & 0 \end{bmatrix}^T, \tag{2.4}$$

for the two dimensional case, where R is the receiver location matrix ($\mathcal{I} \times 3$) and S is the source location matrix ($\mathcal{J} \times 3$) The TOA is defined as

$$t_{ij}^{(r)} = \tau_j^{(s)} + \frac{d_{ij}}{c} - \delta_i^{(r)}. \tag{2.5}$$

Note that the skew is not present. If the event j is also received by a microphone k the two can be subtracted to obtain

$$t_{ij}^{(r)} - t_{kj}^{(r)} = \frac{d_{ij} - d_{kj}}{c} - \delta_i^{(r)} + \delta_k^{(r)}. \tag{2.6}$$

Next [30] states that if this is done two times for the same pair of phones with only changing the sending phone

$$t_{ii}^{(r)} - t_{ij}^{(r)} - t_{ji}^{(r)} + t_{jj}^{(r)} = \frac{d_{ii} - d_{ij} - d_{ji} + d_{jj}}{c} \tag{2.7}$$

is obtained. Since d_{ii}, d_{jj} and c are known; the sum $\frac{d_{ij}+d_{ji}}{c}$ is obtained. Last they try to obtain the relative locations of the microphones, from this distance sums. They use (2.4) to reduce the number of free model parameters and then use an iterative trust-region-reflective algorithm to obtain an estimate of the locations of both the microphones and the loudspeakers. This results in an underdetermined system of equations. A underdetermined system either has no solution or infinitely many solutions. An underdetermined system can be solved in practice if a good initial guess can be made.

2.2.3 Direction of Arrival

The idea behind DOA, sometimes called Angle of Arrival, is that the source location can be determined by looking at the DOA seen from multiple viewpoints. If the relative location of the viewpoints is known the position of the source can also be determined. In general two or more microphones on each device are needed to estimate the DOA. From that moment on algorithms based on Multiple Signal Classification (MUSIC) [31] and Maximum Likelihood (ML) [32, 33, 34] can be used to find an estimate of the loudspeaker locations. This directly gives problem one, we are looking for the microphone locations and not the loudspeaker locations. Another problem is that this method needs a very good quality signal, which is recorded completely synchronous since even small errors in the angle can lead to a big error at some distance. For the Android phones used it is not possible to use two microphones at the same time and even if it is possible to use to microphones it is not possible to tell how the sampling is done.

2.2.4 Received Signal Strength

The last method on localisation that will be briefly explained uses the fact that the power of the signal spreads in multiple directions when travelling from the source to the receiver. Since the relation between amount of spread and the travelled distance is known localisation is possible. If Wi-Fi is used to determine the position, the errors lay between 1 and 5 meters. While the best error with audio is the 22 cm, here the maximum distance is only a couple of meters, depending on the loudness of the source. The research with audio signals is done in [35], they obtained the error of 22 cm. In [36] a study has been done on the practical side of the method. They conclude that it can be an alternative to GPS but only in some environments and indoors is not one of them, so not useful for our purposes.

2.3 Location estimation

After the distances are estimated, the last step is to estimate the locations from the estimated distances. For this there are two methods that can be used to solve our problem of finding the microphone locations. The first one, CMS, is straightforward and finds the best embedding of distance measurements in a least squares sense into a lower-dimensional subspace [37].

The other method is used when a (squared) distance matrix is obtained like in [25]. Here a method from [7] is used to extract the locations. This method uses the SVD to obtain the locations. First the SVD is taken of the squared distance matrix X to obtain

$$RS^T = X = U\Sigma V, \tag{2.8}$$

where R is the receiver location matrix ($\mathcal{I} \times 3$) and S is the source location matrix ($\mathcal{J} \times 3$).

By introducing a new (3 x 3) matrix C, R and S can be extracted into

$$R = UC \tag{2.9a}$$
$$S^T = C^{-1}\Sigma V. \tag{2.9b}$$

To obtain the matrix C a minimization problem has to be solved. To be able to solve this the position of the first microphone has to be fixed, next to this the position of the first source has to be defined as a point on the x-axis. From this point on a gradient decent can be used to find a minimum. In total there are eight minima, one true solution and seven spectral reflections of this solution. The complexity of this optimization is fixed to $\mathcal{O}(9)$ since only nine values have to be found. This value does not change if extra devices are added.

2.4 Conclusion

From the literature survey pressented in this chapter we can concluded that there are a lot of existing localisation methods of which a few can be used for the estimation of the distances.

The distances can be used to find the locations of the microphones. The methods that can be used have two problems: The algorithms are relatively slow because of the time needed to do send the necessary signals and/or due to the slow convergence to the localisation solution. The other problem is that some of the methods also need a minimum of five devices to be able to localisation (3D). In this thesis we want to find a method that is fast and works even when one only wants to localise two devices. We also have seen that most methods do not mention the clock skew, we will try to find out if clock skew may be disregarded or not.

Chapter 3

Time of arrival estimation

In Chapter 1 we have introduced the problems that have to be solved to localize a set of distributed microphones. In order to localize the microphones we first need to obtain accurate TOA estimates In this chapter we will see how the TOA can be estimated.

For localisation algorithms to work property, an accurate TOA measurement is needed. The correct TOA is determined by the direct path between the loudspeaker and microphone, i.e. the distance between them. This property is used for localisation. If there is an error in the TOA, there will also be an error in localisation. It can happen that errors cancel out. For instance, this occurs when TDOA is used and all the errors in TOA estimation are the same. If the estimated TOA is 1/1,000s off, the error can already be 34 cm. In this chapter we will go in depth into the detection of the TOA. First, two different calibration signals will be compared. Second, simulation examples will be provided to show what can be expected. Finally, we draw conclusions about TOA estimation.

3.1 Types of calibration signals

The first problem that is encountered is to find out from which device the signal originates. A straightforward solution is to emit one signal at the time. This is very time consuming since there has to be enough time between the signals to be able to separate them. The better option is to use a different signal for each device. For this to work we need to be able to distinguish the different signals even if there is overlap.

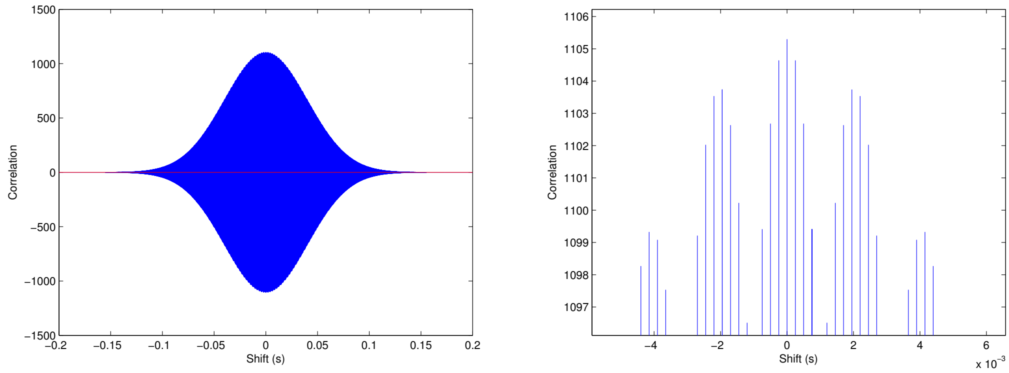
Which signals can be best distinguished depends on the method of detection. Since the original signal is known a matched filter can be used for the detection. A matched filter is the optimal linear filter which correlates the known signal with the recording to detect the presence of the signal [38]. Because correlation is used, signals with a high auto-correlation and a low cross-correlation are best distinguished. Next to the high auto-correlation it is also desirable that it only has one peak at full correlation and is relatively low everywhere else. If this is not the case problems will occur when multiple paths arrive close together. Since time and bandwidth are limited the signals need to be localised in both time and frequency.

3.1.1 Gaussian

The first synchronisation pulse we will discuss is the Gaussian shaped pulse. For this a shape of a Gaussian distribution is modulated with a sinusoidal. The maximum of the pulse in normalized to one, so the envelope is given by

$$G = e^{-a^2}. \tag{3.1}$$

The support of the pulse ranges from $a = -2.5$ to $a = 2.5$ to make sure that values at the edge are close to zero. The length of the pulse is determined by taking $N = \frac{t_{pulse}}{f_s}$ samples in the range of a, where N is rounded to the nearest integer. By modulating this Gaussian shape with sinusoids of different frequencies, different signals can be obtained for the different phones. This means that it is possible to see from which phone the signal originated. Furthermore, the modulated Gaussian pulse is localized in both time and frequency as required. There is only one catch, the auto-correlation of the Gaussian signal is also a sine modulated version of the original Gaussian shape, this means there is not one clear peak that has to be chosen. As can be seen in Fig. 3.1. The secondary peaks are 11 samples of, the third highest are 84 samples of, this gives, in distance, an error of 8.6 cm and 65 cm.



(a) (b)

Fig. 3.1: Auto- (blue) and cross-correlation (red) of two sine modulated Gaussian shapes. The auto-correlation is obtained by from a Gaussian shape modulated with a sine of 10 kHz. For the cross-correlation the Gaussian shape modulated a sine of 10 kHz was correlated with a Gaussian shape modulated with a sine of 20 kHz. The length of the pulse is 0.2 seconds. (a) Whole correlation, (b) Correlation around a shift of 0 seconds

Because of this auto-correlation, it is not possible to accurately distinguish between the main lobe and secondary lobes, especially when there is a channel with more than one path. In other words when there is reverbereation. This can be reduced by choosing a shorter pulse, but then loudness get less, meaning a smaller distance at which the signal still gets detected. Since the path length of the reflections can almost be as long as the path of the direct pulse, the signal has to be very short, less than 0.05 seconds. This to avoid that the signals start to add up resulting in a maximum peak at a different location. If this happens it is impossible to say which peak should be used.

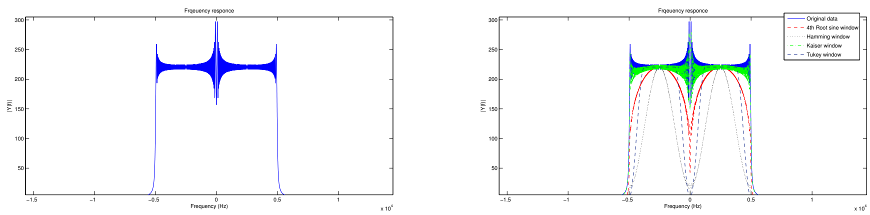
3.1.2 Chirp

The other calibration signal we looked at is the chirp. With a chirp, or sweep, the signal starts at one frequency and stops at another. The transition can be done linearly, but also by other smooth functions as, a second order function. The rate of the transition can be chosen, and thus the length of the signal. So it is a time and frequency localised function as was required. The linear case can be described as

$$x(t) = \sin\left(2\pi t\left(f_0 + f_\delta t\right)\right), \tag{3.2}$$

where $x(t)$ is the output at time t and f_0 is the starting frequency and f_δ is the transition rate.

If the auto-correlation of the chirp is compared to the auto-correlation of the Gaussian shaped pulse there is a big difference. The chirp gives one small main lobe and very small (only 20%) secondary lobes. The frequency response on the other hand is not smooth at all and due to the sudden onset and offset in playback distortions appear in the signal. This causes the auto-correlation to be far worse than what is possible in theory. See Fig. 3.2a for the frequency response of the chirp.



(a) (b)

Fig. 3.2: (a) The original frequency response of a chirp. (b) Frequency response after applying different windows

To remove the distortions caused by the sudden onset and offset in playback a window can be used. Unfortunately most windows like the Hamming cause the secondary lobe to rise. The rise of the secondary lobes is because now only a small range of frequencies are fully present and the rest are suppressed, see Fig. 3.2b. To solve this a window is needed that is more or less flat on top and has a steep sudden unset, unlike most commonly used windows. In accordance with [39] a fourth root sine function is used as window. This window has very steep sides and good suppression of the high and low frequencies, see Fig. 3.3.

One also has to be careful when choosing the frequency ranges. If they lie close together or even overlap the cross-correlation is not as low as it can be, especially when the signals are short. This gives problems at relatively long distances. In that case the cross-correlation with the unwanted signal originating from the recording phone is as strong as the auto-correlation with the wanted signal originating from another phone. This especially happens when there is a little to no reverberation. In practice this means that the frequencies have to be chosen further apart, or a longer signal should be used in order to make sure that there always is an auto-correlation peak that is higher than the cross-correlation peak. That choosing frequencies that lay further apart works is easy to see, but also longer signals do the trick. This is because this increases the auto-correlation more than the cross-correlation. The only parts that correlate are the parts that lay close together in the frequency space, since this is only a small part of the whole signal the increase in in cross-correlation is less than the increase in auto-correlation.

In Sections 3.3 there will be a couple of experiments to see how good the estimation of the

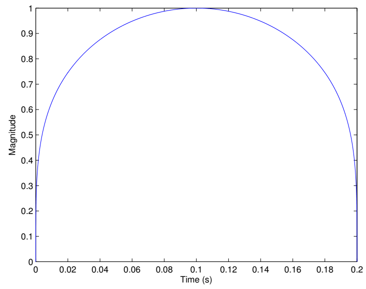


Fig. 3.3: Fourth root sine window

time of arrival is for the different signals, but first we have a look at an example.

3.2 Example

We now have two calibration signals that can be used to find the correct TOA, but as said in Section 1.1.1 we actually estimate a specific channel. By applying a matched filter with one of the calibrations signals as reference one of the channels to the recording phone is obtained. From the channel that is obtained it is possible to retrieve the position of the first peak, if there is a line of sight. This is very easy if there is no noise, then one can really see one clear peak. Recall that noise is defined as all but the signal originating from the direct path. In practice this is not that simple, due to reflections and noise also other peaks appear. Some peaks do not originate from reflections or noise but from the other phones that did send out a signal and in particular the receiving phone that sends out a signal. Although the signal used makes sure this signal is suppressed as much as possible, it is still a signal with a lot of energy since it is close, while the wanted signal can be a signal of low energy.

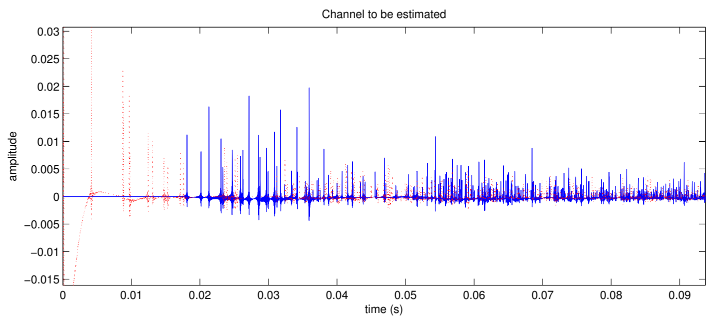


Fig. 3.4: The true channel to be estimated by a chirp and a Gaussian signal

In Fig. 3.4 one sees an example of a channel. Here a red dotted line is added for the signal arriving from the phone itself and a blue one from a phone some distance away. Of course the red line has a peak that is much higher than the other, the task is to estimate both, and for this the calibration signals are used. One can see that the peaks can be close to each other. This can give problems when the auto-correlation has high values over a longer range as is the case whit the Gaussian shaped signal.

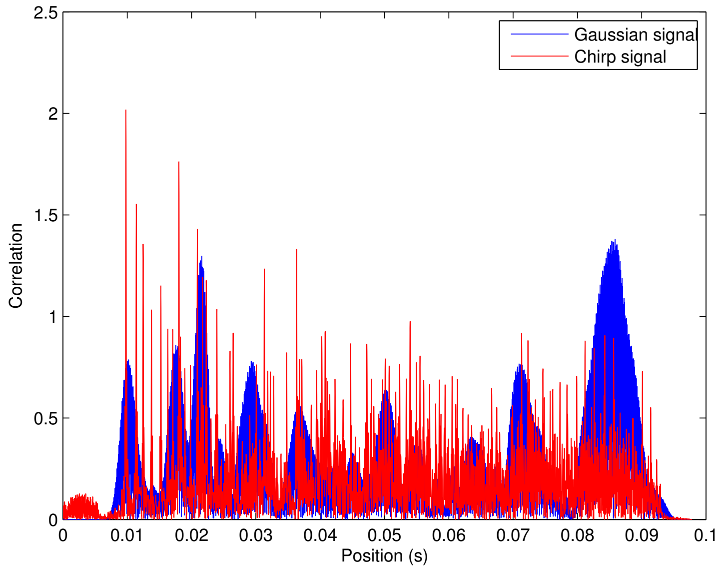


Fig. 3.5: The channel estimated by a chirp and a Gaussian modulated signal. The signal length was 0.025 seconds for both signals.

If the channel is estimated with a Gaussian signal and a chirp the response will be as depicted in Fig. 3.5. For the chirp a signal of 0.025s long starting a 0 Hz to 5 kHz and from 5 kHz to 10 kHz was used. For the Gaussian shaped signals, a Gaussian shape was modulated with sinusoids with a frequency of 1 kHz and 21 kHz, this signal length also was 0.025s. For the Gaussian an ideal case scenario was used, normally the signal would be of chosen to be longer, since this signal now only has a little bit of energy. One can see right away that the chirp has much clearer peaks. The task at hand is to find the first peak, because this is the direct path, and the length of this path is what gives the distance. This is why a good distance between the peaks and the noise floor is needed. One also sees some noise in front of the chirp, this is due to the cross-correlation that does not suppresses all of the unwanted signal, especially since the signal is so short in this example. In both cases one sees that the first peak is somewhere just around 0.01 seconds, if one zooms in into this region Fig. 3.6 will be obtained.

In this figure the difference between the two methods is really visible. With the Gaussian signal one can really see that the secondary lobes are as high as the primary lobe. Due to correlation, that has large values over a long range, the signals originating from the different paths start to add up. In this case the peak form the second path disappears into the first, causing the first maximum to shift. This means that the incorrect peak is picked if the largest peak is chosen, and thus an error in localisation.

If we compare the Gaussian shaped signal and the chirp we see that the major differences are in the auto- and cross-correlation. The Gaussian shaped signal has a lower cross-correlation, while the chirp has a narrower auto-correlation. Because the chirp is less prone to errors we decide to use the chirp in the rest of this thesis.

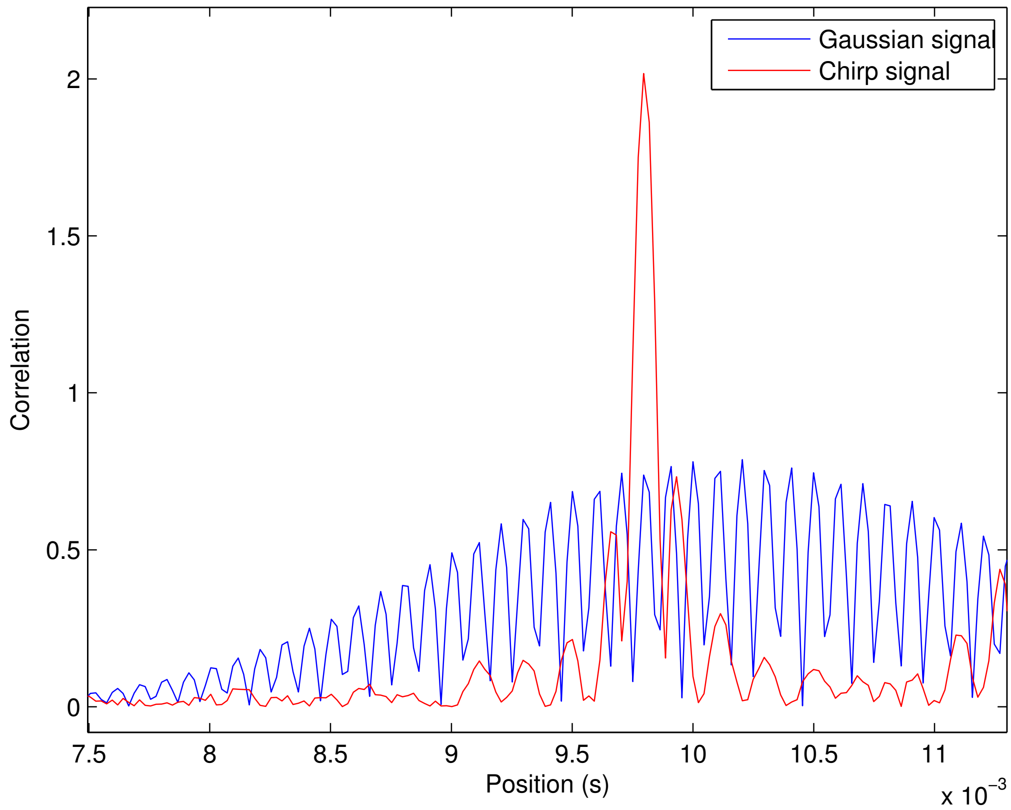


Fig. 3.6: The channel estimated by two types of signals, zoomed in on the first lobe

3.3 Simulations

To see where any errors occur in the estimation of the TOA three set-ups where used. In all set-ups we looked at the error at different distances between two phones. With the first experiment the amount of noise was varied, for the second the length of the pulses and for the last one we varied the reverberation time. Each experiment was repeated 20 times and the mean error was taken for the figures. This resulted in Figs. 3.7, 3.8 and 3.9.

The error in the figures is calculated as follows: first the difference with the true TOA was calculated for both sides, only for the signal originating from the other phone (the weaker signal). The absolute value was taken of these two values and after this they were added to each other, which is defined as the error. From this the mean was calculated for the figures. Since the values could be large, 1000 samples or more, the values where truncated to 50 samples. This means an error of more than 39 cm.

Fig. 3.7 shows the effects of noise on the estimation of the TOA. It is clear that as long as the noise is below a threshold it is good and after this it is not detectable. This means that noise does not shift the pulses but makes detection impossible. This is a good thing because now we only need to be below some threshold and not in an environment with the least amount of

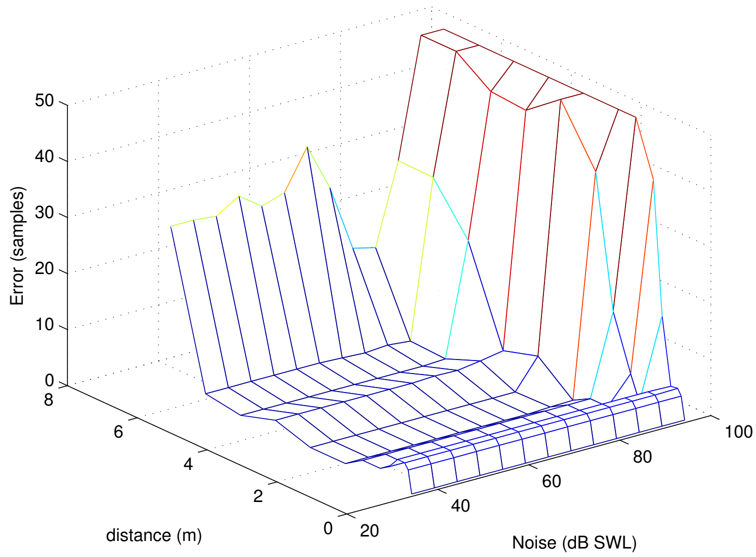


Fig. 3.7: The effects of noise on the estimation of the TOA, with a reverberation time of 0.4 seconds and a signal length of 0.2 seconds. Mean error in samples over 20 experiments, truncated at 50.

noise possible.

In Fig. 3.8 the effects of the signal length on the TOA estimation can be found. Here it is clear that there is a minimum length for the calibration signal. A length of more than 0.15 seconds seems to be good choose. Although the error decreases if longer pulses are used, it may not way up against the costs. Since the improvement is neglect-able compared to the increase in signal length. We decide to use a signal length of 0.2 seconds for all other experiments.

In the last figure of this experiment, Fig. 3.9 the effects of the reverberation time on the TOA estimation are made clear. The interesting thing is that a low amount of reverberation gives the same error as a large amount of reverberation.

Note that there is an error up to 2 m in all the results. What causes this error is still unknown. Manual peak picking results in the same answer so detection itself is not the problem and since the maximum error is only 5 samples it was left this way for now. The error that occurs at 7 meters is because the signal falls below our detection threshold. This threshold can be adjusted with as consequence that it will be more vulnerable to noise when trying to find the peak originating from the direct path.

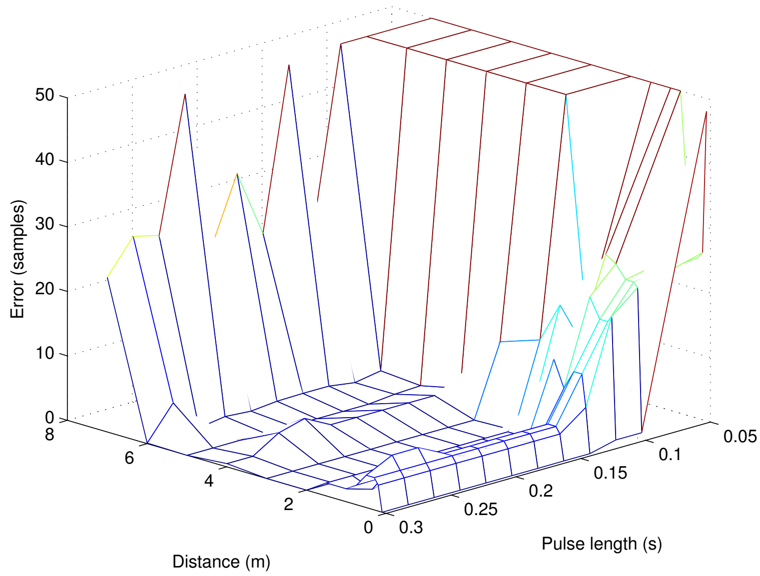


Fig. 3.8: The effects of the signal length on the estimation of the TOA, with a reverberation time of 0.4 seconds and a noise level of 65 dB SWL. Mean error in samples over 20 experiments, truncated at 50.

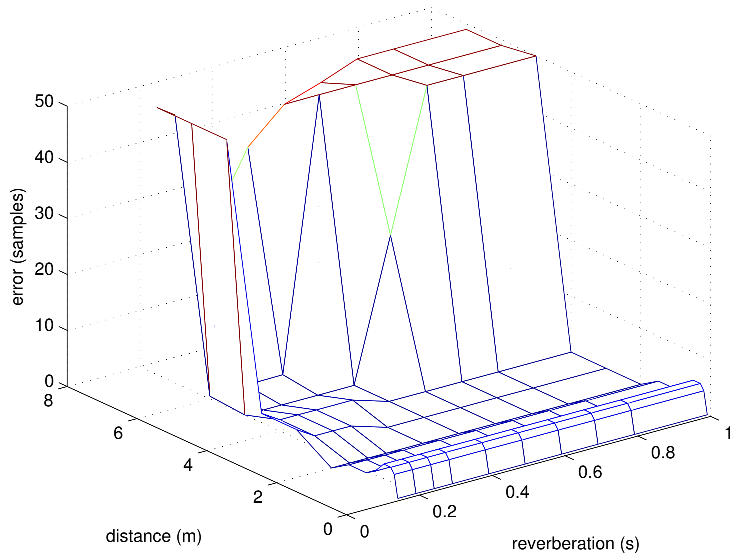


Fig. 3.9: The effects of reverberation on the estimation of the TOA, with a noise level of 65 dB SWL and a signal length of 0.2 seconds. Mean error in samples over 20 experiments, truncated at 50.

3.4 Conclusion

We have looked at two kinds of calibration signals. Both the Gaussian shaped signal and the chirp can work. The Gaussian needs to be really short to be able to distinguish the different paths of the channel. If we want to use the chirp we have to apply a window to remove the onset effects. The fourth root sine window works good, while more commonly known windows lose the sharp clear peak. Because the cross-correlation of the chirp is not as low as that of the Gaussian, one has to take care of not picking the cross-correlation peak at larger distances. We decided to use the chirp signal with a signal length of 0.2 seconds for the rest of this thesis.

Chapter 4

Time synchronization and internal delays

In Chapter 3 we have seen how to estimate the TOA that is needed to solve the localisation problem that was introduced in Chapter 1. Next to the TOA we also need to estimate the clock parameters. In this chapter we will see how these parameters can be estimated and thus create one ”common” clock.

4.1 Skew estimation (Δ)

To gain insight in the dissipation of time according to the devices, i.e. insight in the skew parameter, we need to look at time differences between measurements. The crystal oscillators in the phones that determine to clock speed are different for different phones and even more so for different brands of phones. They also depend on temperature so in a sense also on the amount of processes running on the phone. Since temperature is a slowly varying process the clock speed can be assumed constant over a small amount of time, for at least a few minutes.

One way to estimate the skew is to send two calibration signals after each other. If the second pulse is sent an exact amount $T_{int} = \tau_w(t_1) - \tau_w(t_0)$ later, the signal should also arrive this amount later, any differences are due to the skew. With (1.7) we can calculate the skew by dividing the measured time difference by T_{int}. But this only works if there is a method to send a pulse an exact amount of time later. Since this is in general not possible, another solution has to be found.

As we have seen in Chapter 2 one of the options is not to compute the actual skew but the relative skew. If the assumption is made that the skew is only due to the difference in clock speeds, the relative skew can be computed. Since the recording and playback are buffered processes, they do not (directly) depend on the amount of other running processes. This is true as long as the buffer does not run out of space, the assumption now is made that this does not happen. This is a reasonable assumption since the buffer size is normally chosen to prevent this from happening.

Say that skew directly influences the sampling rates, thus that there is one speed for the AD[1] and DA[2] converter. This would imply that a skew higher than one results is a higher frequency signal at the sending side. This is because more samples are send in the same amount of time. At the recording side sampling is also done at this higher rate, so it looks like the frequency is lower than it actually was, because more samples are received in the same amount of time. As long as the two skews are the same the skews cannot be calculated, because the two effects

[1]Analogue to Digital
[2]Digital to Analogue

counter each other. But as they differ it is possible to say how they relate to each other. This is done in exactly the same way, but since the exact time difference is not known the relative skew is obtained. In practice this can achieved by sending two or more calibration signals in one audio file, so two signals are sent at the following moments:

$$\tau_1^{(s)} = \frac{t_j^{(s)} + \epsilon_i^{(s)}}{\Delta_i} \tag{4.1a}$$

$$\tau_2^{(s)} = \frac{t_j^{(s)} + \epsilon_i^{(s)}}{\Delta_i} + \frac{T_{int}}{\Delta_i}, \tag{4.1b}$$

where $\tau_x^{(s)}$ is the actual time pulse number x left the loudspeaker, $t_j^{(s)}$ is the time the send command is given, $\epsilon^{(s)}$ is noise due to internal delays and T_{int} is the pulse interval.

If (4.1) is combined with (1.1) and (1.9b) to

$$t_{ij,1}^{(r)} = \left(\frac{t_j^{(s)} + \epsilon_j^{(s)}}{\Delta_j} + \frac{d_{ij}}{c} \right) \Delta_i + \delta_i^{(r)} + q_{ij,1}^{(r)} \tag{4.2a}$$

$$t_{ij,2}^{(r)} = \left(\frac{t_j^{(s)} + \epsilon_j^{(s)} + T_{int}}{\Delta_j} + \frac{d_{ij}}{c} \right) \Delta_i + \delta_i^{(r)} + q_{ij,2}^{(r)}, \tag{4.2b}$$

the moment the calibration signals are registered is obtained. $t_{ij,x}^{(r)}$ is the time event x send from phone j is registered on phone i and $q_{ij,x}^{(r)}$ is the sampling noise of event x. Because the sampling noise is only a very small quantity we will neglected it from this point on.

On all the phones the difference between the calibration signals can be calculated to obtain

$$\bar{t}_{ij} = T_{int} \frac{\Delta_i}{\Delta_j}, \tag{4.3}$$

where $\bar{t}_{ij} = t_{ij,2}^{(r)} - t_{ij,1}^{(r)}$. By dividing $\bar{t}_{ij}$ by T_{int} an estimate of $\frac{\Delta_i}{\Delta_j}$ is obtained.

If the crystal oscillator of the phones would be very accurate and known, it would not be necessary to calculate the clock skew, but this is not the case as can be seen in Fig. 4.1. To make this figure a measurement of 60 seconds was done, whereby a Samsung phone sent out a chirp of 0.2s every 0.25s. The sampling rate of the phones was set to 44100 Hz. The error was measured in number of samples difference.

Fig. 4.1 gives us some valuable insight. The first thing one might notice is that there is a lot of fluctuation in the offset in the beginning of the measurement. Here one sees only one figure, but if you look at all 12 measurements, the beginning always contains this fluctuation. (see appendix B) After about 30 calibration signals, so after 7-8 seconds, it clears up. Since all the recordings of all the phones show the same fluctuation this is error is likely to come from the playback side, although we cannot be sure about this. We can see that the difference in offset

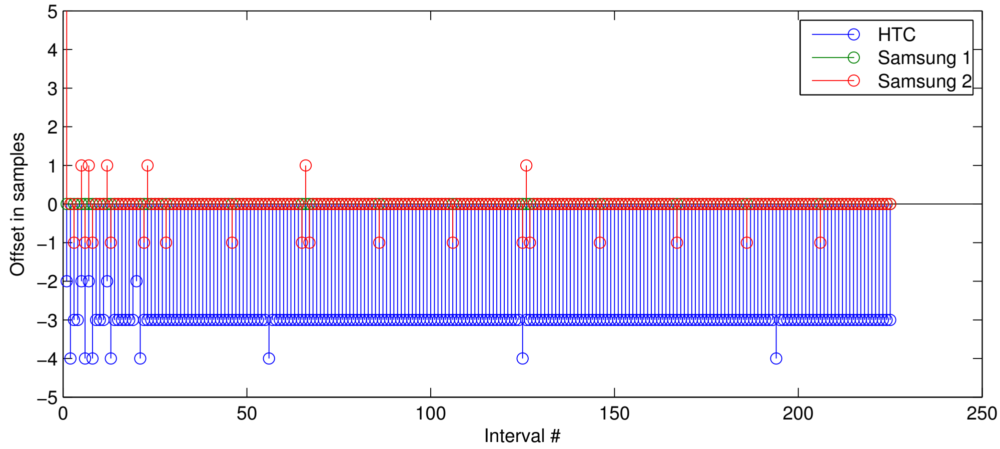


Fig. 4.1: Progression of the offset between phones. Two Samsung's (Galaxy SII) and one HTC (Sensation). To make this figure a measurement of 60 seconds was done, whereby a Samsung phone sent out a chirp of 0.2s every 0.25s. The sampling rate of the phones was set to 44100 Hz. The error was measured in number of samples difference.

is compensated on the next sample, with the exception of some samples that are really of. Those are due to the selection of the incorrect TOA. Looking at the rest of the recoding we see periodic spikes with a different period for the different phones. These spikes are introduced by the sampling of the signal. If it would be possible to record a continuous signal there would not be any spikes but the actual differences between the clocks. This also happens by averaging over an infinite long recording, this is essence the same as what is done with RBS [12].

In Table 4.1 we find the results of all the measurements. What can be learned from this table is that the skew is not completely stable, although the average skew is always the same the spike interval changes from measured to measurement and thus there is a small change in the skew. The interval in the table is the average interval of the recording. This is why slight changes are certainly possible but changes up to 12.5s are not to be appointed on doing an averaging over a relatively short measurement. But the estimated skew can be used for some time. When this measurement was repeated one month later the average skew was again exactly the same for the different pairs. The spike interval had an average offset of 2.5s with respect to a measurement one month earlier. This means that if a rough estimation is good enough, this estimation can be done once and then use it for longer period of time, say a couple of months. But for accurate measurements it is better to do it each time.

Another interesting thing about the skew estimation is that it matters which phone is sending. One would only expect a sign change in both the average and the interval if the sending and receiving phone change, but this is not the case. In some cases it is as expected with the Samsung 2 and HTC for instance. With the HTC sending we have an average of minus one every 8.5s, and the other way around the average is plus one every 8.9s. But if we look to Samsung 1 and the HTC it is a completely different story. With the HTC sending the average is plus one every 12.75s and the other way around its minus one every 18s.

Let us take the average difference of three samples between each calibration signal that is obtained if the HTC is involved. This means that the offset grows with three samples every

Table 4.1: Table with clock skew estimation. In the column skew you find the skew w.r.t. the sending phone. In the last three columns you will find the interval at which an extra difference in TOA is added (or subtracted).

	Skew			HTC		Samsung 1		Samsung 2	
#	HTC	Samsung 1	Samsung 2	Action	Interval (s)	Action	Interval (s)	Action	Interval (s)
1	0	3	3	-		+1	13.5	-1	8.1
2	0	3	3	-		+1	12.9	-1	8.5
3	0	3	3	-		+1	12.8	-1	8.6
4	0	3	3	-		+1	12.3	-1	8.8
5	-3	0	0	-1	17.3	-		-1	4.9
6	-3	0	0	-1	17.8	-		-1	5.0
7	-3	0	0	-1	18.3	-		-1	5.1
8	-3	0	0	-1	18.8	-		-1	5.0
9	-3	0	0	+1	10.5	+1	0.0	-	
10	-3	0	0	+1	9.8	+1	0.0	-	
11	-3	0	0	+1	8.0	+1	12.5	-	
12	-3	0	0	+1	7.4	+1	9.8	-	

0.25 seconds. If the different calibration signals are sent one second after each other the relative skew results in an error of 12 samples which equals an error of 10 centimetres. This means that the skew cannot be neglected for longer measurements. Because of this we will incorporate the skew.

4.2 Offset estimation (δ)

In Chapter 1 we defined the offset with (1.6). This offset is defined with respect to a real-time standard. We can also define the relative offset, this would give

$$\delta = \tau_{p1}(t_0) - \tau_{p2}(t_0), \tag{4.4}$$

where $\tau_{p1}(t_0)$ is the time according phone 1 at t_0 and $\tau_{p2}(t_0)$ the time according to phone 2 at t_0. We also need to define a t_0. We define it as the beginning of the recording. This can be completely different moments if it is defined in a real-time standard. But by doing this we do not need to save the start time of the recording, the recording itself is all that is needed.

To be able to calculate the relative offset between phones, one has to know the distances between one loudspeaker and the microphones of two or more devices. The relative offset will be calculated of the recording devices. It is best if the distances are equal else compensation is needed but at least the distances must be small, later on one can see why this is. The relative offset now is defined as the difference between the estimated TOAs. If the distances are not equal the extra travel time has to be subtracted from the one with the extra distance, but this will give an error due to the skews that are not known ($t = \frac{d_1\Delta_1 - d_2\Delta_2}{c}$).

Equations (1.1) and (1.9b) can be combined to

$$t_{ij}^{(r)} = \left(\tau_j^{(s)} + \frac{d_{ij}}{c}\right)\Delta_i + \delta_i^{(r)} + \epsilon_i^{(r)}, \tag{4.5}$$

where q is again neglected.

If we rewrite the different skews into one skew k and put all the knowns to one side, we get

$$\left(t_{ij}^{(r)} - \frac{d_{ij}}{c}\right)\frac{\Delta_k}{\Delta_i} = \tau_j^{(s)}\Delta_k + \left(\delta_i^{(r)} + \epsilon_i^{(r)}\right)\frac{\Delta_k}{\Delta_i} + (1-\Delta_k)\,\frac{d_{ij}}{c}. \tag{4.6}$$

Note, although the relation between the skews is known the actual values are unknown.

A skew of one would mean a perfect clock. The clock of an Android phone can run of its own without synchronisation for more than a few hours being seconds of. This means that the skew of an Android phone is close to one. Since it can be assumed that the skew always has a value that is very close to one, and since the distances between the phones are small, $(1-\Delta_k)\,\frac{d_{ij}}{c}$ can be neglected. This will give a small error but it will be far smaller than one sample, so not noticeable.

Let us define a new variable z_x that is equal to $\delta_x^{(r)} + \epsilon_x^{(r)}$. When (4.6) is put into a matrix,

$$A = \begin{bmatrix} \tau_1^{(s)}\Delta_k + z_1\frac{\Delta_k}{\Delta_1} & \tau_2^{(s)}\Delta_k + z_1\frac{\Delta_k}{\Delta_1} & \dots & \tau_I^{(s)}\Delta_k + z_1\frac{\Delta_k}{\Delta_1} \\ \tau_1^{(s)}\Delta_k + z_2\frac{\Delta_k}{\Delta_2} & \tau_2^{(s)}\Delta_k + z_2\frac{\Delta_k}{\Delta_2} & \dots & \tau_I^{(s)}\Delta_k + z_2\frac{\Delta_k}{\Delta_2} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_1^{(s)}\Delta_k + z_I\frac{\Delta_k}{\Delta_I} & \tau_2^{(s)}\Delta_k + z_I\frac{\Delta_k}{\Delta_I} & \dots & \tau_I^{(s)}\Delta_k + z_I\frac{\Delta_k}{\Delta_I} \end{bmatrix} \tag{4.7}$$

is obtained.

Next subtract the first row from every other row, to

$$B = \begin{bmatrix} z_2\frac{\Delta_k}{\Delta_2} - z_1\frac{\Delta_k}{\Delta_1} & z_2\frac{\Delta_k}{\Delta_2} - z_1\frac{\Delta_k}{\Delta_1} & \dots & z_2\frac{\Delta_k}{\Delta_2} - z_1\frac{\Delta_k}{\Delta_1} \\ z_3\frac{\Delta_k}{\Delta_3} - z_1\frac{\Delta_k}{\Delta_1} & z_3\frac{\Delta_k}{\Delta_3} - z_1\frac{\Delta_k}{\Delta_1} & \dots & z_3\frac{\Delta_k}{\Delta_3} - z_1\frac{\Delta_k}{\Delta_1} \\ \vdots & \vdots & \ddots & \vdots \\ z_I\frac{\Delta_k}{\Delta_I} - z_1\frac{\Delta_k}{\Delta_1} & z_I\frac{\Delta_k}{\Delta_I} - z_1\frac{\Delta_k}{\Delta_1} & \dots & z_I\frac{\Delta_k}{\Delta_I} - z_1\frac{\Delta_k}{\Delta_1} \end{bmatrix}. \tag{4.8}$$

Here we end up with the relative offsets that depend on the relative skew. The actual offsets can only be obtained when one offset, z_1, is known. If this is not known we end up with an error depending on the magnitude of the relative skew and z_x. Besides this the neglecting of the sampling noise gives an additional error.

4.3 Conclusion

In this chapter we have seen that the relative clock skew can be estimated. The relative offset cannot be estimated because of the clock skew. Next to this we have not jet obtained a method to estimate ϵ_j^s. This parameter can be avoided if a TDOA method is used.

In the end we want a fast method to localize Android phones. First estimating the clock parameters and then do localisation will not result in a fast method. Because the following needs to be done:

1. Position the Android phones with care.
2. Determine the exact distances between the microphones.
3. Estimate the relative offset and skew of the phones.
4. Reposition the phones to their wanted locations.
5. Do a TDOA based measurement.
6. Calculate the relative locations of the phones.

This is a lot of work and takes time so not a fast method. The relative clock skew can be calculated this way since only one phone has to send a second calibration signal just after the first, which takes not even halve a second. Since [30] has found a way to overcome the estimation of the offset we will extend there method with the estimating of the skew and see if this improves there estimation.

Chapter 5

TOA based localisation and Results

In the previous chapters we have defined the localisation problem. This problem we have partly solved. In Chapter 3 we have found that a chirp can be used to estimate the correct TOA and in Chapter 4 we defined a method to estimate the relative clock skew and the offset. We have also seen that (direct) estimation the offset is time consuming and should be avoided for a fast localisation algorithm.

In this chapter a method to estimate the distance between two phones will be updated so that it uses the relative clock skew. Once all distances are known it is relatively easy to determine the relative locations. The method that is used is based upon [30], the difference is in the estimation of the skew and the use of different calibration signals for the different phones. First this updated method will be explained. Then we confirm that this improves the existing method by using both simulated and real-data experiments.

5.1 Method

Since the method is based on [30], which already is explained in Chapter 2, only the differences will be discussed here. The fundamentals of this method for estimating the distance between two phones is shown in the example of Fig. 1.1.

We start with (4.5) which is reproduced here for the convenience of the reader.

$$t_{ij}^{(r)} = \left(\tau_j^{(s)} + \frac{d_{ij}}{c}\right)\Delta_i + \delta_i^{(r)} + \epsilon_i^{(r)}.$$

Note that this relation only looks at the recording side. The actual time the pulse left the speakers is used in the equation, not the time the command to send was given. As discussed before it is not possible to estimate the time between the moment the command was given and the moment signal starts to propagate. This is why one can just use this time as the send time.

Of this moment the skew is unknown, to solve this, another pulse has to be send. This is done as is described in Section 4.1, this way the relative skew, $\frac{\Delta_i}{\Delta_j}$, is obtained.

The relative skew can now be combined with (4.5) to

$$t_{ij}^{(r)}\frac{\Delta_k}{\Delta_i} = \left(\tau_j^{(s)} + \frac{d_{ij}}{c}\right)\Delta_k + \delta_i^{(r)}\frac{\Delta_k}{\Delta_i} + \epsilon_i^{(r)}\frac{\Delta_k}{\Delta_i}, \tag{5.1}$$

where k is denoting one of the phones.

For easy notation a new variable is introduced, $\hat{t}_{ij}^{(r,k)}$ that equals

$$\hat{t}_{ij}^{(r,k)} = t_{ij}^{(r)} \frac{\Delta_k}{\Delta_i}. \tag{5.2}$$

From here on we can continue with Hennecke et al.[30]. They apply (2.5), (2.6) and (2.7). The measured value in all these equations was t we apply the same equation with our counterpart of t, $\hat{t}$, this results in

$$\hat{t}_{ii}^{(r,k)} + \hat{t}_{jj}^{(r,k)} - \hat{t}_{ij}^{(r,k)} - \hat{t}_{ji}^{(r,k)} = \Delta_k \left(\frac{d_{ii} + d_{jj}}{c} - \frac{d_{ij} + d_{ji}}{c} \right). \tag{5.3}$$

From this point on only one phone with a known skew is needed to obtain the $\frac{d_{ii}+d_{jj}}{c} - \frac{d_{ij}+d_{ji}}{c}$ that [30] uses. Since d_{ii} and d_{jj} are known $\frac{d_{ij}+d_{ji}}{c}$ is obtained. The distance between the microphone of phones i and j is now defined as $\frac{d_{ij}+d_{ji}}{2}$, this distance is used to determine the relative location of the microphones using MDS. If there is no phone of which the skew is known an error is obtained when Δ_k is neglected. The size of this error will only depend on the Δ_k. As we have seen in Chapter 4 this is not the case if the skew is neglected and not all TODs are the same.

5.2 Results

This section is divided in two parts. The first part will focus on the results of the computer simulations that where preformed. The second part will consist of results gained while preforming real-data experiments with actual phones. From the real-data experiments it is not possible to verify all parameters, but any differences between the computer simulations and real-data experiments will be made clear. It was also possible to see if there was any improvement in error with our extension on the work of [30].

5.2.1 Computer simulated results

Two computer simulations where preformed: distance estimation and actual localisation. For the distance-estimation only two phones where used. Here the goal was to see on what range this method would work under various conditions. For the localisation part two or more phones where used. Here the goal was to see what would happen if the localisation was preformed with multiple phones at once under various conditions.

Distance estimation

For the first experiment we looked at what noise would do to the localization error. The amount noise that was added to the signal is defined in dB SPL. To get some idea of the values, 65 dB compares to a conversation at one meter, 80 dB to the sound in a bus, 95 dB to the sound of a passing metro and 110 dB to a loud car horn at one meter. The same experiment was preformed to look at the error in the estimation of the TOA. The error is expected to follow the shame shape. The experiment was repeated 20 times the mean error is used. The measurement was preformed with a reverberation time of 0.4 seconds. The length of the calibration signal was 0.2 seconds. In Fig. 5.1 one sees the results of this experiment. As expected there is noise threshold. Below this threshold localisation works. If we compare this figure with Fig. 3.7 we see that this error indeed originates in the TOA estimation.

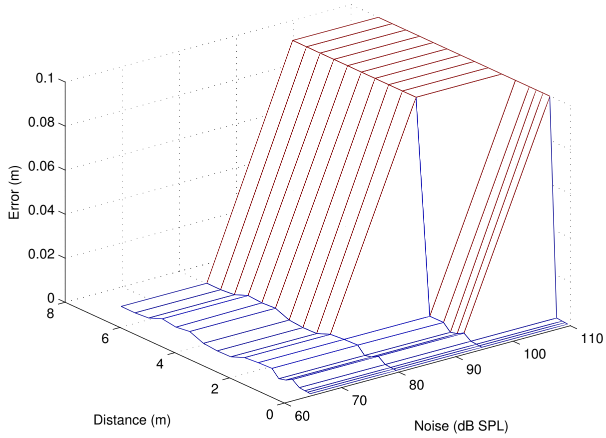


Fig. 5.1: The effects of noise to the maximum distance. The mean error over 20 experiments in meters, measured with a pulse length of 0.2 seconds and a reverberation of 0.4 seconds. The error is truncated at 10 centimetres.

The following simultion investigated the effects of reverberation on the maximum distance. For this a pulse of length 0.5 seconds was used and a noise level of 55 dB SPL. This length and noise level where used to make sure the error was due to the reverberation and not some other effect. The best result would be very stable and always give the same numbers. This would mean that reverberation has no effect on our algorithm. In Fig. 5.2 one sees the results of this experiment. The effect due to the reverberation makes that the maximum distance decreases. If we compare this figure with Fig. 3.9 we see that this error, as the error due to the noise, originates in the TOA estimation as is expected.

The final computer simulation only involving distances was to see what happens if we rotate the phones around there axes. Since there is some distance between the loudspeaker and microphone, this angle will influence the error. The expectation is that the error will have a maximum of $\frac{d_{ii}+d_{jj}}{4}$. This will happen when both the microphones and both the loudspeakers lay on one line, but one phone is rotated 180 degrees with respect to the other. The expectation is that the minimum error will be less than 3.5 mm, since this is half a sample offset. This is the very best we can do with this sampling frequency (44.100 Hz). The angle of the phones is given with respect to the line through both microphones. The position of the microphones is not changed during the experiment, but the positions of the loudspeakers are changed. For the experiment the distance between the phones was 2 meters, with a reverberation of 0.4

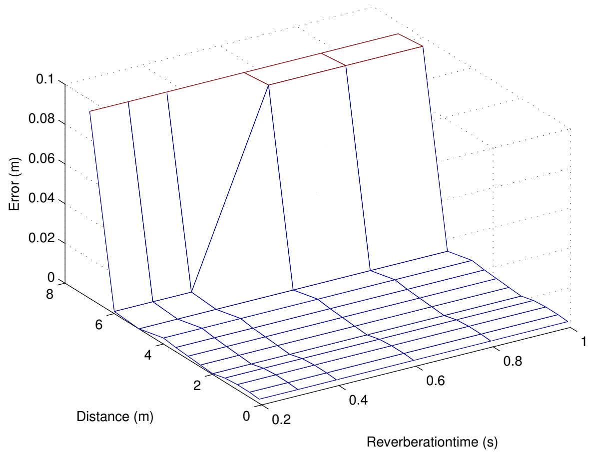


Fig. 5.2: The effects of reverberation to the maximum distance. The mean error over 20 experiments in meters, measured with a pulse length of 0.5 seconds and a noise level of 55 dB SPL.The error is truncated at 10 centimetres.

seconds, a pulse length of 0.2 seconds and a noise level of 55 dB SPL.

Fig. 5.3 clearly shows two maximum errors. The errors occur when the angle of one phone is 0 degrees and the other angle is 180 degrees. The clear minimum error occurs when the angles are equal, as expected. This also happens if the two angles add up to 360 degrees. The maximum error is also as we expected, $\frac{d_{ii}+d_{jj}}{4}$. If we look to the distance that is obtained by looking to the points $\frac{r_i - s_i}{2}$ and $\frac{r_j - s_j}{2}$, Fig. 5.4 is obtained. Here the error is always close to zero, this means that the distance between those points is actually measured with the proposed method.

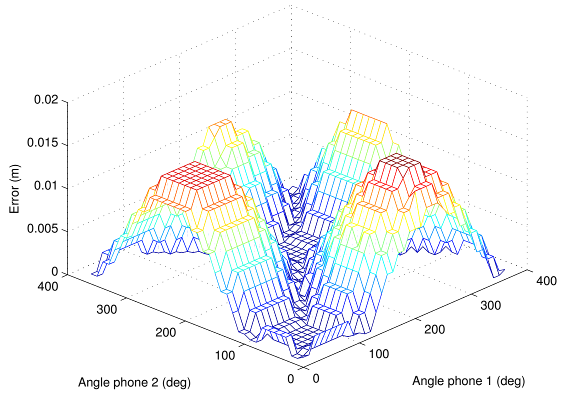


Fig. 5.3: The effects of rotation on the error when looking to the microphone location. For the experiments we used: a microphone distance of 2 meters, a reverberation time of 0.4 seconds, a pulse length of 0.2 seconds and a noise level of 55 dB SPL. Mean error over 20 measurements.

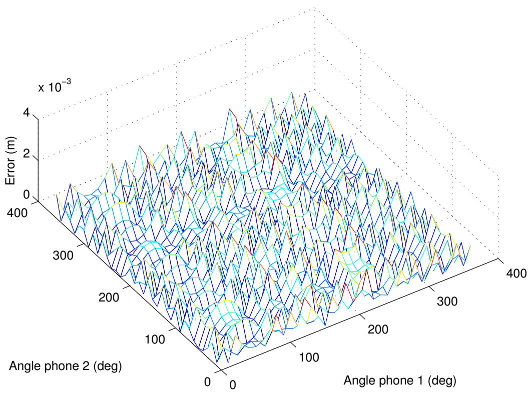


Fig. 5.4: The effects of rotation on the error when looking to the points $\frac{r_i - s_i}{2}$ and $\frac{r_j - s_j}{2}$. For the experiments we used: a microphones distance of 2 meters, a reverberation of 0.4 seconds, a pulse length of 0.2 seconds and a noise level of 55 dB SPL. Mean error over 20 measurements.

Localization

For the localisation part of the computer simulations the number of phones was increased from 2 to 10 in order to investigate the effects on localization accuracy for increasing number of devices. Because the calibration signals where sent out within some interval some the pulses arrive at the same instance and thus cause interference. The available bandwidth was split amongst the phones $BW = \frac{22}{I}$kHz. We wanted to see if errors would reduce or increase with more phones. Since one only needs to know the location of four phones to position all other phones, it could be that the error starts to reduce with more than four phones. But on the other hand, less bandwidth and more interference would make the TOA estimation more difficult, and thus cause more errors.

In Fig. 5.5 the obtained error is visualized. For up to five phones there is only a very small error after this a jump in the error occurs and then it stabilizes again. It is clear that adding extra phones results in a rising error and not in a better estimate.

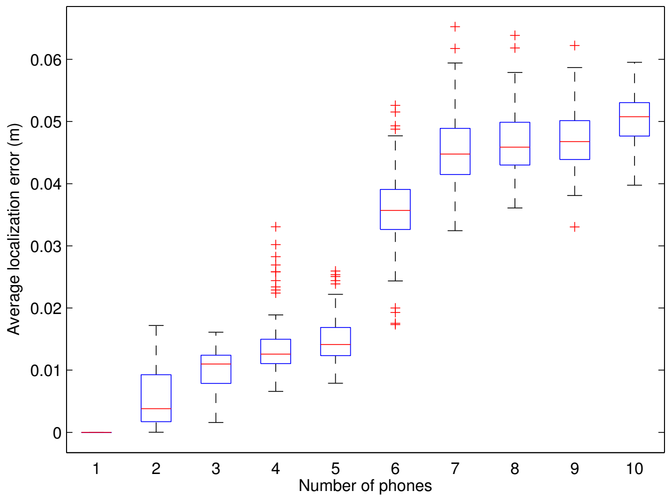


Fig. 5.5: The effect on the error when adding extra phones, 0.4 seconds reverberation, 65 dB SPL, pulse length 0.2s and 100 random realizations within an area of 4 by 5 meters for each number of phones.

5.2.2 Real-data experimental results

For the real-data experiments two Samsung's (Galaxy SII) and one HTC (Sensation) where used. Two types of experiments where preformed: distance estimation and localisation. The first experiment was preformed with only the two Samsung phones. The second was preformed with all three phones.

Distance estimation

For the distance estimation two phones were put on a table (0.75 x 1.50) and the distance between the microphones was increased from 10 cm to 1.25 m. At each location the experiment was repeated six times. The phones where put side by side. This means that the loudspeaker and microphone where under a small angle (2 to 5 degrees), the angle was the same for both of the phones. The results can be found in Fig. 5.6

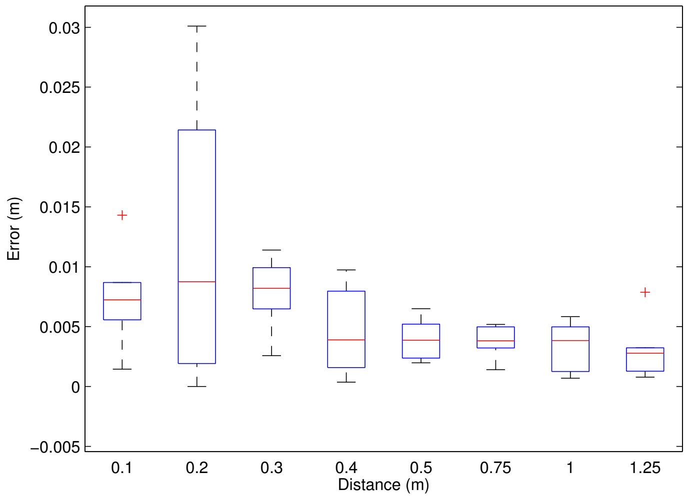


Fig. 5.6: Error in the estimated distance between two Samsung phones for eight distances repeated six times each.

The error that is found is a little bit higher than the error obtained in the computer simulations. This can be explained by a small positioning error in the real-data experiments. This is because we do not exactly know the location of the microphones. Given the set-up used an measurement inaccuracy of two millimetres can be assumed in the location of each of the phones. With this assumption the error is in the same range and thus for this measurement simulations can be used.

The distance-estimation experiment is also preformed in [30], they obtained the average error over 5 experiments. There errors are: 2 mm at 10 cm, 10 mm at 20 cm, 35 mm at 30 cm and 60 mm at 40 cm. Thus there error at 10 cm is a bit smaller and the rest of the errors is bigger. Furthermore they obtain an error that increases with the distance, while the error we obtained remains stable, this is probably due to the skew that is considered in our case. As said this occurs when there is a time difference between the TODs. Since only one synchronisation pulse is used by Hennecke et al.the time between the pulses has to be big enough to visually separate them. This can result in errors of this magnitude.

Localization

The last experiment preformed was the localisation of the three phones. For this 7 different constellations of the phones where used, see Fig. 5.7. For each constellation the experiment was repeated 10 times. From the result a box plot was made, which is shown in Fig. 5.8.

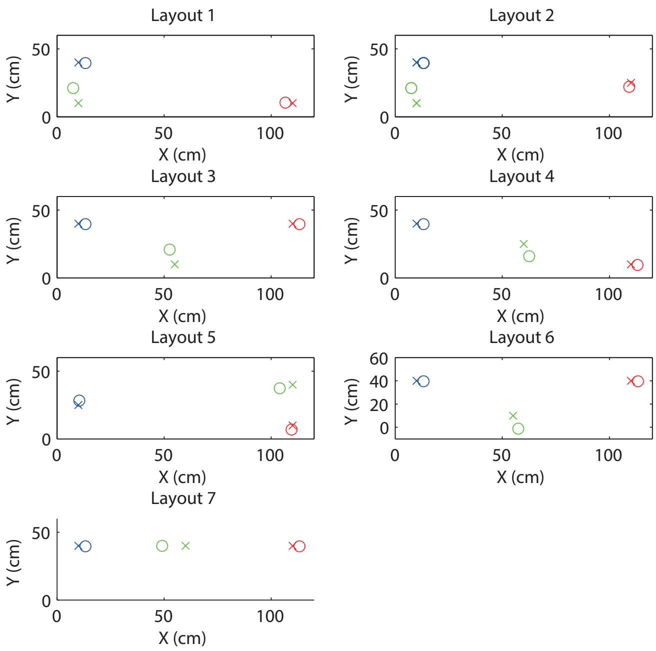


Fig. 5.7: The 7 different layouts. The x's denote the microphone location the o's denote the loudspeaker location. The green phone is the HTC this one has a microphone loudspeaker distance of 10 cm.

As can be seen in this figure the error is quite large for some orientations and quite small for others. This is because of the large distance between the microphone and loudspeaker on the HTC phone. Two of the three distances can be off by as far as 6.5 centimetres ($\frac{10+3}{2}$). This is what is the case with layout number 2. As the HTC phone is turned in a more favourable position, layout number 5, this error is greatly reduced.

One can really see what the effect of the different angles is (Between the HTC and the Samsungs). Recall that the angle is always defined with respect to the line through the two microphones. Only two angles can be compared at the same time unless more than two microphones lay on one line as is the case in layout 7.

If the sum of the different angles is big the error is big. If the sum is small the error is small. This effect is bigger when the HTC phone is involved because of the larger distance between the microphone and loudspeaker.

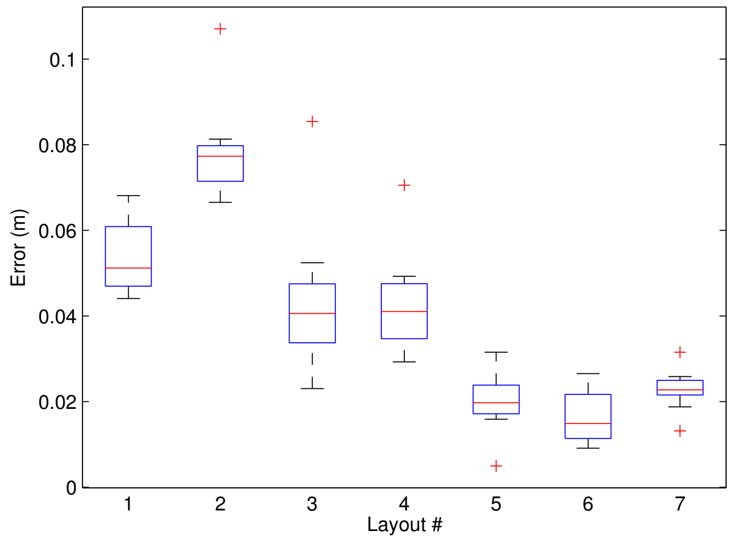


Fig. 5.8: The error in the localisation of three phones for different layouts.

5.3 Conclusions

We tried to improve the existing algorithm by including a skew estimation, here we have proven a theoretical maximum error that directly depends on the distance between the microphone and loudspeaker on the phones. From the computer simulations we can conclude that the error rises when more phones are added. We also have seen that the results from the real-data experiments are comparable to the computer simulations. If we compare our real-data results with the real-data results of [30] we find that our method gives an error that, to some extent, does not depend on the distance between the phones. Where the error in [30] rapidly raises to 6 cm at a distance between the phones of 40 cm while the error from our extention remains at less than 1 cm even at distances of more than 1 m. Unfortunately the angle of the phones he used is not defined, nor the brand and type of phones so it is hard to compare the actual results.

If more than 7-8 phones are used the method from [25] would be preferable. This is because this method gives better results with an increasing amount of phones. However this method might also be improved by adding the skew.

The error in localisation highly depends on the orientation of the phones together with the distance between the microphone and loudspeaker on the phones. If this distance is small enough good localisation is possible with our extension. On the other hand as this distance becomes quite large, as is the case with the HTC phone, the error highly depends on the angle of this phone compared to the angle of the other phones. The size of the error can be predicted if the difference between the angles is known, but it is not possible to say if the estimated distance is too small or too large.

Chapter 6

Recommendations and Conclusions

In this chapter we first focus on an outlook for future work. This will be topics that showed up too late to really look into, and topics that where too large to do a real in depth study. After this the conclusions of this thesis can be found.

6.1 Recommendations

In this section the focus will be on topics within this thesis that might need some additional attention to improve it future. The topics are ordered according there appearance in the thesis. So we start with improvements that can be made on the software used.

6.1.1 Software

In the current implementation of the localization software we first initiate the recording and tha calibrations signals are emitted some time later. This is done to make sure that all phones have started recording. The time between the start of the recording and the first signal leaving the loudspeaker can be long in some cases and (too) short in others. An improvement would be to implement a feedback machanism which assures that all phones aree recording before the calibrations signal is transmitted. This in itself is not that difficult. The problem here is that one somehow needs to send a command back to the phones, which is faster to arrive then messages via the Google servers. This can be done because from the moment the phones respond to say they started recording, one can use other methods than the Google servers to contact the phones. This is since the phones can also send back other information, for instance its IP-address so further messages can be directly send over the Wi-Fi network.

6.1.2 Pulse shape

For the pulses two kinds of pulses have been discussed, both have advantages and disadvantages. In the end we chose the chirp because of its autocorrelation properties, but there might be different kinds of pluses with good auto- and cross-correlation properties. Since any type of pulse that has a low cross-correlation and a high auto-correlation with an output that looks like a delta pulse could be used one might look into this further and see if there are better signals.

6.1.3 Localization

The actual localisation is now done on a portable computer, this is fine but it would be far better if this could be done on the phones itself. Since the main part of the localisation is the application of a matched filter and finding the direct path of the channel it might be perfectly fine to do this on the phones itself. This way everybody can set-up the system at all times, without the need of some additional hardware.

It might also be a good idea to look if localisation can be improved by means of the obtained channel. Since a good estimate of the channel is obtained for at least the first, and a part of the second order reflections. This is extra information on the locations of the microphones and loudspeakers, but also includes extra unknowns. Assuming a rectangular room, the locations of the walls, ceiling and floor can be given by six distances. This means that in theory it is possible to retrieve the actual locations of the microphones and loudspeakers of two phones with 4 extra TOAs:

$$3\,(\mathcal{I}+\mathcal{J})+6 \leq n\mathcal{I}\mathcal{J} => n \geq 5 \;\; for \; \mathcal{I}=\mathcal{J}=2$$

The biggest problem here probably is what pulse came from which reflection. If this question can be somehow solved, there can be a serious improvement on the current error.

It might also be possible to do some kind of validation of the obtained distance with RSS based estimation of the distance. Since RSS does not depend on the estimation of the TOA it is a method that can be used to validate the estimated TOA. Now it sometimes occurs that an error of a few meters is obtained because of a wrongly picked TOA. RSS is not very accurate, but the error of 22 cm [35] that can be obtained is small enough to validate the TOA.

6.2 Conclusions

We have seen that estimating the TOA is actually the estimation of the channel if this can be done the TOA can also be estimated. With the Gaussian shaped calibration signal the channel cannot be estimated when multiple paths are received close together. We have seen that the channel can be estimated when a chip signal is used. The problem of the chirp is that its cross-correlation is not as well suppressed as the cross-correlation of the Gaussian shaped signal. This gives problems at larger distances, in other words when the wanted signal is very weak. To solve this more bandwidth or a longer signal should be used but this means that the localisation takes longer.

By sending out two pulses in one audio file we can estimate the relative skew. By implementing this into the existing algorithm of [30], we have seen the localisation error can be decreased. From the computer simulations we can conclude that this method not excites an average error of 5 cm until 10 phones are used. We have also seen that the correct distance can be estimated up to a distance of 6 meters between two phones.

The error in localisation highly depends on the orientation of the phones and the distance between the microphone and loudspeaker on the phones. If this distance is small enough good localisation is possible with our extension.

When the proposed method is implemented in a real-world scenario, the main improvement is in the stability of the error at increasing distances. At a distance of 40 cm our extension causes the error to drop by 6 cm compared to [30]. Since the improvement we have booked on this algorithm, by including the skew, our expectation is that the including of the skew will also improve other methods that use Android phones and TOA based localisation.

Appendix A

Software

The software used to make the recordings for localisation consists of two parts. The first part is an Android program on the phones. The second part is a PHP script on an Apache web server. Both will be explained.

A.1 Android application

If the Android application is installed it should just work, there is no real configuration for normal use. The calibration signals should be stored in the folder "/pulses/". With the menu button on the phone you can register the phone at the preconfigured server or unregister it. If the phone is registered it can receive commands from the Apache server. The code itself consists of five classes: one class controls the sending of pulses, one class controls the recording, one class controls the uploading of files, one class listens for new commands, and one class to combine all others.

For the communication the Google cloud Messaging is used. To be able to use this a Google API account is needed. In the settings of this account one can specify which servers can send messages to the registered phones. The phones need some ID that belongs to the account, this can also be found on the web page with the settings. On this moment this service is free and an unlimited amount of messages can be sent.

A.2 Web server

For the web server Apache was used together with PHP, because we have experience in this language. There are three main scripts: one to register and unregister the phones from the server, one the send settings to the phones and one to receive files from the phones. Since this is a normal web server MATLAB® can communicate with the server by using the commands `urlread()` and `urlwrite()`, `urlread()` returns a string with the contends of the requested page, while `urlwrite()` writes the contends of the page to a specified file on your computer.

A.2.1 (Un)Registering

When registering the server gets two identifier strings, one random string that is used by the Google servers to send a message to the phone and one to identify the phone. This last one is used make sure the same phone can only register once. When unregistering the same information is supplied to remove the phone from the list of known phones.

A.2.2 Configuring phones

For the configuration of the phones 6 commands can be send to the phones, namely: setting settings, start recording, stop recording, start sending, stop sending and upload.

Set Settings This is the largest command, here the individual phone settings can be set. The following settings are available:

foldername	The name of the folder where the file should be stored
filename	The name of the file where the recording should be stored
pulsename	The name of the file that contains the calibration signal, this file is played whole a can consists of more than one signal
startime	The amount of time to wait before sending starts
repeattime	The interval at widths sending should repeat
number of pulses	The number of times to send a calibration signal

The *filename*, *pulsename* and *starttime* can be set individually for each phone while the other settings will be the same for all the phones. All setting are always send to all phones at the same time to keep it simple.

Start recording This command starts the recording on the phones. It first makes the folder and the file to sore it in. After this the real recording starts. This command is always send to all phones at once and it does not influences the playback status.

Stop recording This command stops the recording on the phones. It first stops the recording and then waits until the buffer is empty, once this is the case it sends the file to the server. The phones receive the server location once they receive the settings. It does not influences the playback status.

Start sending This command starts the sending of the calibration signals by playing a WAVE file at some interval until stopped or it runs out of pulses to send.

Stop sending This command stops the sending of the calibration signals and resets the number of pulses left to send.

Upload This command requests a particularly file from the server, normally the files are already send when the recording stops, but sometimes this fails, with this command the file can be requested again.

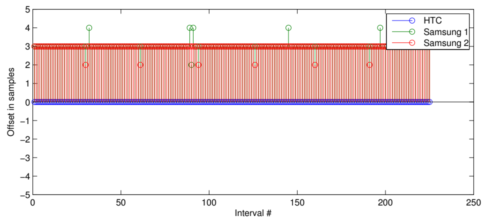
A.2.3 Receiving files

When a file is send to the server it is stored in the same folder as it was on the phone, the file name is also the same. Since the file name is different for each phones multiple files end up in the same folder on the server. This files can later be downloaded by MATLAB® for this the `urlwrite()` command is needed, since this command writes the requested contend to a file on your computer.

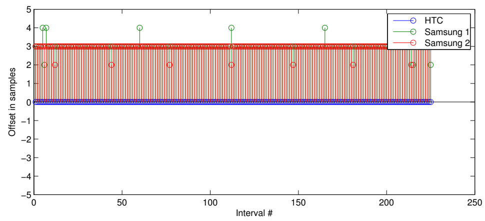
Appendix B

Skew images

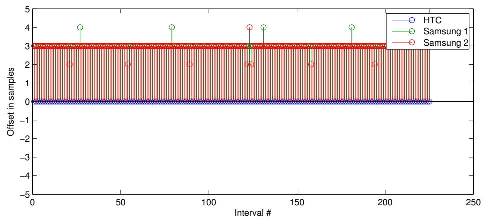
Here one finds an overview of all the measurements done for the skew estimation. This measurements where done width two Samsung's (Galaxy SII) and one HTC (Sensation). For each figure a measurement of 60 seconds done, whereby one of the phones did send out a chirp of 0.2s every 0.25s. The sampling rate of the phones was set to 44100 Hz. The error was measured in number of samples.



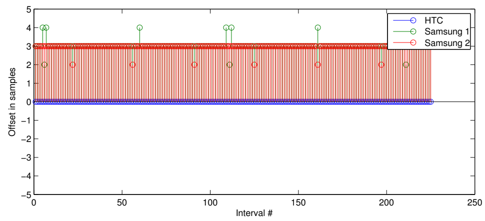
(a) Sending HTC, measurement 1



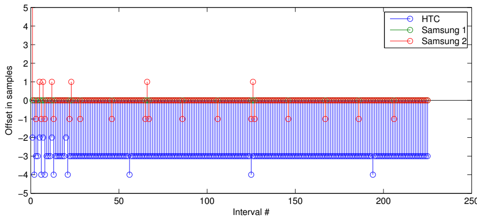
(b) Sending HTC, measurement 2



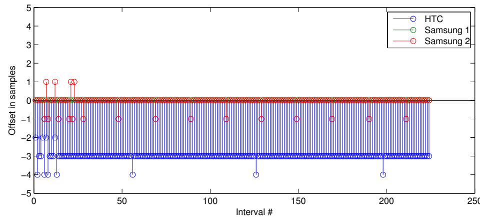
(c) Sending HTC, measurement 3



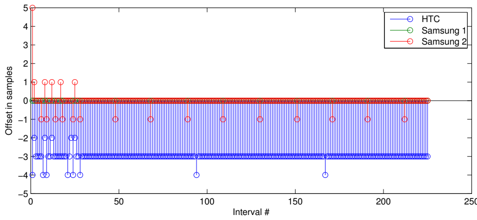
(d) Sending HTC, measurement 4



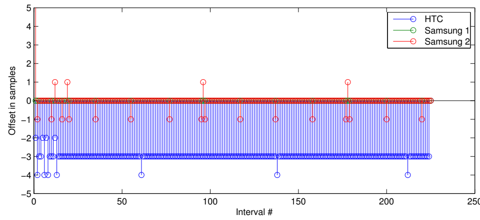
(e) Sending Samsung 1, measurement 1



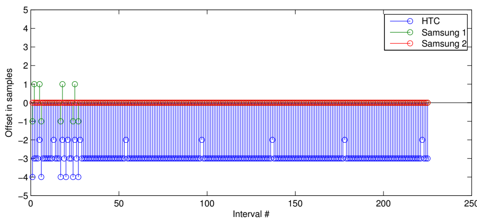
(f) Sending Samsung 1, measurement 2



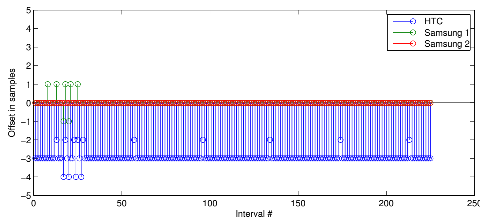
(g) Sending Samsung 1, measurement 3



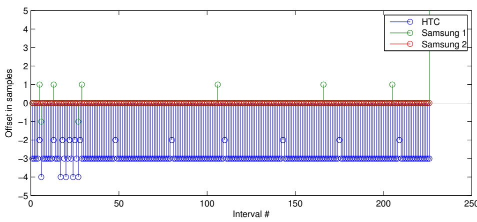
(h) Sending Samsung 1, measurement 4



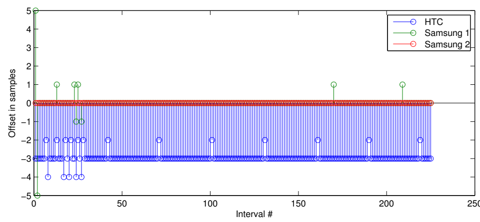
(i) Sending Samsung 2, measurement 1



(j) Sending Samsung 2, measurement 2



(k) Sending Samsung 2, measurement 3



(l) Sending Samsung 2, measurement 4

Bibliography

[1] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '79.*, vol. 4, 1979, pp. 208–211.

[2] J. Ortega-García and J. González-Rodríguez, "Overview of speech enhancement techniques for automatic speaker recognition," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 2. IEEE, 1996, pp. 929–932.

[3] R. C. Hendriks, T. Gerkmann, and J. Jensen, "DFT-Domain Based Single-Microphone Noise Reduction for Speech Enhancement: A Survey of the State of the Art," *Synthesis Lectures on Speech and Audio Processing*, vol. 9, no. 1, pp. 1–80, 2013.

[4] B. Widrow, J. R. Glover Jr, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong Jr, and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proc. IEEE*, vol. 63, no. 12, pp. 1692–1716, 1975.

[5] R. Mucci, "A comparison of efficient beamforming algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 3, pp. 548–558, 1984.

[6] I. Lunden. (2013) Android, led by samsung, continues to storm the smartphone market, pushing a global 70http://home.tiscali.nl/ehabets/rirgenerator.html

[7] M. Crocco, A. Del Bue, and V. Murino, "A bilinear approach to the position self-calibration of multiple sensors," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 660–673, 2012.

[8] W. C. Sabine, *Collected papers on acoustics.* Harvard university press, 1922.

[9] E. A. Habets, "Room impulse response generator," *Technische Universiteit Eindhoven, Tech. Rep*, vol. 2, no. 2.4, p. 1, 2006.

[10] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, p. 943, 1979.

[11] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[12] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, Dec. 2002. [Online]. Available: http://doi.acm.org/10.1145/844128.844143

[13] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 37, no. 6, pp. 1067–1080, 2007.

[14] S. Pandey and P. Agrawal, "A survey on localization techniques for wireless networks," *Journal of the Chinese Institute of Engineers*, vol. 29, no. 7, pp. 1125–1148, 2006.

[15] U. Larsson, J. Forsberg, and A. Wernersson, "Mobile robot localization: integrating measurements from a time-of-flight laser," *IEEE Trans. Ind. Electron.*, vol. 43, no. 3, pp. 422–431, 1996.

[16] S. Lanzisera, D. T. Lin, and K. S. Pister, "Rf time of flight ranging for wireless sensor network localization," in *Intelligent Solutions in Embedded Systems, 2006 International Workshop on.* IEEE, 2006, pp. 1–12.

[17] S. Gezici and Z. Sahinoglu, "Uwb geolocation techniques for ieee 802.15. 4a personal area networks," *Mitsubishi Electric Research Laboratory Technical Report TR-2004-110*, 2004.

[18] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 41–53, 2005.

[19] J. J. Caffery and G. L. Stuber, "Overview of radiolocation in cdma cellular systems," *IEEE Commun. Mag.*, vol. 36, no. 4, pp. 38–45, 1998.

[20] J. Caffery Jr and G. L. Stuber, "Subscriber location in cdma cellular networks," *IEEE Trans. Veh. Technol.*, vol. 47, no. 2, pp. 406–416, 1998.

[21] Y.-T. Chan, H. Yau Chin Hang, and P.-C. Ching, "Exact and approximate maximum likelihood localization algorithms," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 10–16, 2006.

[22] J. J. Caffery Jr, "A new approach to the geometry of toa location," in *Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd*, vol. 4. IEEE, 2000, pp. 1943–1949.

[23] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on.* IEEE, 2005, pp. 91–98.

[24] I. Guvenc, S. Gezici, F. Watanabe, and H. Inamura, "Enhancements to linear least squares localization through reference selection and ml estimation," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE.* IEEE, 2008, pp. 284–289.

[25] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, "Autolocalization in ad-hoc microphone arrays," in *Proc. ICASSP 2013, Vancoucer, Canada, May 26-31 2013*, pp. 106–110.

[26] J. Ta, "Global positioning system," Ph.D. dissertation, California State University, 2011.

[27] S. T. Birchfield and A. Subramanya, "Microphone array position calibration by basis-point classical multidimensional scaling," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 1025–1034, 2005.

[28] Y. Chan and K. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 1905–1915, 1994.

[29] V. C. Raykar, I. V. Kozintsev, and R. Lienhart, "Position calibration of microphones and loudspeakers in distributed computing platforms," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1, pp. 70–83, 2005.

[30] M. H. Hennecke and G. A. Fink, "Towards acoustic self-localization of ad hoc smartphone arrays," in *Hands-free Speech Communication and Microphone Arrays (HSCMA), 2011 Joint Workshop on.* IEEE, 2011, pp. 127–132.

[31] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, 1986.

[32] F. Schweppe, "Sensor-array data processing for multiple-signal sources," *IEEE Trans. Inf. Theory*, vol. 14, no. 2, pp. 294–305, 1968.

[33] I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 10, pp. 1553–1560, 1988.

[34] J. Chen, R. Hudson, and K. Yao, "Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field," *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1843–1854, 2002.

[35] Z. Liu, Z. Zhang, L.-W. He, and P. Chou, "Energy-based sound source localization and gain normalization for ad hoc microphone arrays," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2. IEEE, 2007, pp. II–761.

[36] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 41–52, 2007.

[37] T. F. Cox and M. A. Cox, *Multidimensional scaling.* CRC Press, 2010.

[38] R. Brunelli and T. Poggiot, "Template matching: Matched spatial filters and beyond," *Pattern Recognition*, vol. 30, no. 5, pp. 751–768, 1997.

[39] D. W. Caress. (2010, Jun.) Some comments on the signal processing used by chipt subbottom profiles. [Online]. Available: http://www.ldeo.columbia.edu/res/pi/MB-System/sonarfunction/SubbottomProcessing/subbottomdataprocessing.html