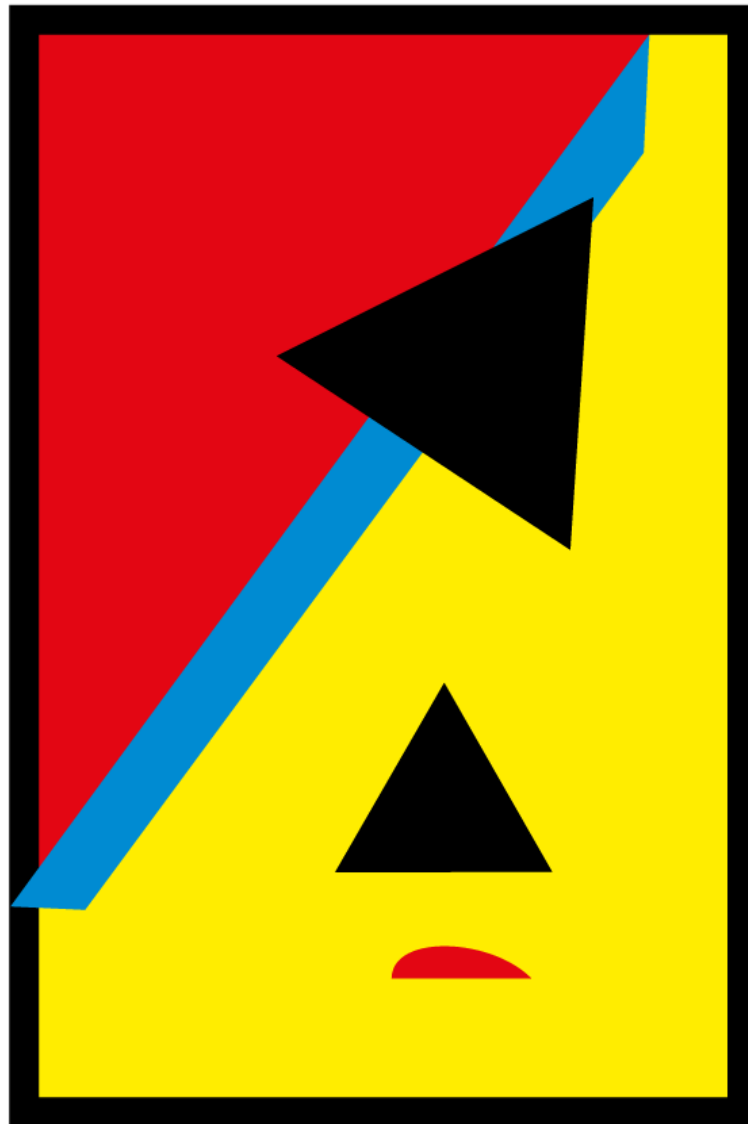


Sound location detection

Afstudeer verslag



ALTEN

Student: Tim Scholten 407225

Onderwijsinstelling: Saxion

Begeleider onderwijsinstelling: Eelco Jannink

Begeleiders ALTEN: Hugo Logmans

1 Samenvatting

ALTEN is geïnteresseerd in de mogelijkheden van locatie bepaling met behulp van geluiddetectoren. Er wordt in dit project onderzoek gedaan naar de verschillende aspecten van locatiebepaling op basis van geluid, en hier zijn een aantal onderzoeksvragen uit voort gekomen. Deze onderzoeksvragen worden onderzocht met literatuuronderzoek, waarna er aan de hand van een Proof of concept een aantal systeemtesten uitgevoerd gaan worden gekeken of de praktijk overeenkomt met de theorie.

De hoofdvraag luidt: *Hoe accuraat is het mogelijk om de realtime locatie van een bewegende geluidsbron te bepalen met behulp van geluiddetectoren?*

Een hypothese is dat de omgeving veel invloed gaat hebben op de metingen, en dat het vooral in afgesloten omgevingen moeizamer zal gaan om consistente meetwaarden te krijgen. De verwachting is dat de precisie van de metingen, naast de omgeving, erg afhankelijk zal zijn van variabelen zoals de afstand tussen microfoons en de gebruikte lokaliseringmethode. De hoop is dat de automatische kalibratie dichtbij genoeg bij de echte waarden van de afstanden tussen de microfoons kan komen dat het geen significante impact zal hebben op de meetresultaten. De lokaliseringmethode die in theorie het meest geschikt zou zijn voor dit project is SRP-PHAT. Door de manier van locatie bepaling, kan dit algoritme robuuste resultaten in onstuimige omgevingen produceren.

Uit de resultaten van de systeemtesten is gebleken dat het mogelijk is om met het SRP-PHAT algoritme, tot op 4cm nauwkeurig de locatie van een geluidsbron te kunnen bepalen. Gemiddeld was de nauwkeurigheid van het systeem, over alle scenario's, is ~1.1 meter. Hierbij bleek dat er op grotere afstanden van de microfoons, een grotere kans op onnauwkeurige metingen.

De resultaten van de kalibratietesten bevestigen dat het meten van consistente resultaten moeizamer is in een gesloten omgeving, vergeleken met een omgeving in de buitenlucht. De kalibratietesten die in de buitenlucht gedaan zijn resulteerde in een gemiddeld procentuele afstandsverschil van 15.9%.

Aanbevolen wordt om meer tests uit te voeren en het proof of concept uit te breiden met identificatie van geluid en/of machine learning.

2 Versiebeheer

Versie	Status	Datum
0.1	Eerste versie, implementatie van huisstijl, structuur en Plan van aanpak hoofdstukken.	20-02-2020
0.2	Bronnen verzameld, structuur gemaakt, ontwerp begonnen, bestaande stukken uitgebreid en aangepast.	12-03-2020
0.3	Hardware ontwerp gemaakt, structuur omgegooid gefocust op eindresultaat, bronnen bij de tekst/kopjes staan waar ze nodig zijn.	23-04-2020
0.4	Feedback van Merijn verwerkt en structuur verbeterd op aanraden van Eelco.	20-05-2020
1.0	Concept verslag inlevering.	02-06-2020
1.1	Eindverslag inlevering.	15-06-2020

Tabel 1 - Versiebeheer

3 Begrippenlijst

Begrip	Beschrijving
POC	Proof of concept
Cross correlation	Het vergelijken van een signalen die ontvangen zijn op meerdere detectoren, om te kijken of hetzelfde geluid gedetecteerd is, en om te bepalen wat het verschil in tijd is tussen het ontvangen van de geluiden. (Bracewell, 1965)
FFT	Fast Fourier Transform (FFT) is een snelle manier om een Discrete Fourier Transformation te doen. Dit is een wiskundige lineaire transformatie die veel gebruikt wordt voor het uitvoeren van digitale signaalverwerking, frequentie analyses en bij het uitvoeren van discrete convoluties. (Proakis & Manolakis, 1996).
Convolutie	Een convolutie is een wiskundige methode die twee functies vergelijkt en een derde functie genereert die uitdrukt hoe de vorm van de twee functies vergelijkbaar zijn. (Smith, 1997)
TDOA	Time difference of arrival. Het verschil in tijd tussen het waarnemen van geluidsgolven op 2 verschillende microfoons. Bijvoorbeeld: microfoon 1 ontvangt het geluid 5 milliseconden eerder dan microfoon 2
Triangulatie	Een driehoeksmeting. Hiermee kan een locatie bepaald worden wanneer 2 hoeken en de afstand tussen deze 2 hoeken bekend zijn.
Gain	De gevoeligheid die soms ingesteld kan worden op een microfoon
Heat map	Een heat map is een datavisualisatie techniek die de significantie als kleur in twee dimensies laat zien
Sample rate	“Bemonsterings snelheid” De hoeveelheid datapunten er per seconde opgehaald kunnen worden op een apparaat van een bepaalde input.
Amp	Een audio versterker
Breakout	Elektrische componenten hebben vaak een geïntegreerd circuit. Deze worden er dan ‘uit gehaald’ met een breakout, zodat er andere elektrische componenten op aangesloten kunnen worden
Ground noise	Wanneer de ground en het signaal van een apparaat over dezelfde draad gaat, ontstaat er ruis
Mic	Kort voor microfoon, vaak gebruikt in combinatie met een getal om aan te duiden welke microfoon er bedoeld wordt. Deze nummering gaat in de context van dit verslag van 0 t/m 2
RMSE	Root Mean Squared Error. De wortel van het gemiddelde van de foutafwijking van een bepaalde meting
Hot-swappable	Een onderdeel van de functionaliteit kunnen uitwisselen voor een vergelijkbare toepassing, zonder de applicatie opnieuw op te starten of af hoeven te sluiten.
Iterable	Een python datatype waar overheen geïtereerd kan worden.
Baud-rate	De maximaal aantal bytes die per seconde over de seriële port van een device gestuurd/ontvangen kunnen worden.
ADC	Analog to digital converter. Converteert een analoog signaal van bijvoorbeeld een sensor naar leesbare data.

Tabel 2 - Begrippenlijst

4 Inhoud

1 Samenvatting	1
2 Versiebeheer.....	2
3 Begrippenlijst	3
4 Inhoud.....	4
5 Inleiding	6
6 Opdracht.....	7
6.1 Aanpak.....	7
6.2 Afbakening	7
6.2.1 Binnen scope	8
6.2.2 Buiten scope.....	8
7 Onderzoeksvragen.....	9
7.1 Hoofdvraag.....	9
7.2 Deelvragen	9
7.2.1 Deelvraag 1: Omgeving.....	10
7.2.2 Deelvraag 2: Lokaliseringsmethode	12
7.2.3 Deelvraag 3: Aanpasbare variabelen	14
7.2.4 Deelvraag 4: Precisie	15
7.2.5 Deelvraag 5: Kalibratie	17
8 Hardware	19
8.1 Microfoons	19
8.2 Kabels	20
8.3 Central monitoring device	22
8.4 Hardware design	23
9 Software	25
9.1 Architectuur.....	25
9.2 Teensy	26
9.2.1 Requirements	26
9.2.2 Taal.....	26
9.2.3 Overzicht.....	27
9.3 Desktop applicatie	29
9.3.1 Requirements	29
9.3.2 Talen	29

9.3.3 Structuur.....	30
9.3.4 Sequence diagrams	32
9.3.5 Performance.....	38
9.3.6 Code conventies	39
9.3.7 Unittesten	39
10 Systeemtesten	40
10.1 Testen uitvoeren.....	40
10.2 Scenario's	40
10.2.1 Scenario 1	40
10.2.2 Scenario's 2 en 3	41
11 Resultaten.....	43
11.1 Automatische kalibratie	43
11.1.1 Scenario 1	43
11.1.2 Scenario 2.....	44
11.1.3 Scenario 3.....	44
11.2 Locatiebepaling.....	44
11.2.1 Scenario 1.....	44
11.2.2 Scenario 2	46
11.2.3 Scenario 3	47
11.3 Onderzoeksvragen	49
12 Conclusie.....	51
13 Aanbevelingen.....	52
14 Nawoord.....	53
15 Bibliografie.....	54
16 Bijlagen	57
16.1 Resultaten overige.....	57
16.2 Unittesten.....	57
16.3 Foto's	58
16.4 Reflectie.....	62

5 Inleiding

Positiebepaling wordt gebruikt voor allerlei doeleinden. Op grote schaal, zoals het tracken van de locatie van iemands mobiel met behulp van gps, tot op kleine schaal, zoals een bewegingsdetector in een kamer om een lamp aan te zetten. ALTEN is geïnteresseerd in locatiebepaling op kleine schaal. Hoe accurater hoe beter. Ze doen onderzoek naar de mogelijkheden in positiebepaling en de technieken die hier allemaal voor gebruikt kunnen worden. Dit project is opgesteld om de mogelijkheden van het gebruik van geluidsdetectoren bij locatiedetectie te ontdekken.

Lokalisering van een geluidsbron, wat komt daar allemaal bij kijken? Welke methoden zijn hiervoor er beschikbaar, en welke is hiervan geschikt? Heeft de omgeving waarin een geluid gemeten wordt nog invloed op de lokalisering ervan? Zo veel vragen waar dit project verder op in gaat, in literatuur en in de praktijk. Nu vraag je je af, is dit allemaal niet al onderzocht?

Het is al door meerdere onderzoeken bewezen dat de locatie van een geluidsbron, tot op zekere hoogte, bepaald kan worden met behulp van meerdere geluidsdetectoren (Hoang Do, 2007) (Maximo Cobos, 2017). Wat de meeste van deze onderzoeken met elkaar gemeen hebben, is dat de locatie van de microfoons in een vaste verhouding ten opzichte van elkaar gepositioneerd zijn. Wat als het mogelijk zou zijn dit een variabele te maken? Vanuit deze gedachte is het idee ontstaan van het maken van een systeem dat automatisch gekalibreerd kan worden en de locatie van een geluidsbron kan bepalen. Hoe accuraat kan een locatie op deze manier bepaald worden? Hoe kan deze automatische kalibratie gedaan worden? Weegt het op tegen handmatige kalibratie? We gaan het zien.

Fun fact: lokalisering en lokalisatie zijn gespeld met een K, maar locatie is gespeld met een C. Dit komt omdat alle woorden die eindigen op 'katie' [kaatsie] veranderd zijn naar 'catie', maar de woorden die hiervan afgeleid zijn, niet. (locatie-lokatie, 2011)

6 Opdracht

ALTEN heeft een opdracht samengesteld om de kennis die aanwezig is binnen het bedrijf over positiebepaling te verbreden. Deze opdracht is gericht op positiebepaling met behulp van geluidsdetectoren.

6.1 Aanpak

Het doel van dit project is om de mogelijkheden te ontdekken omtrent geluidsbron lokalisering. De manier waarop dit aangepakt gaat worden is aan de hand van een onderzoeksfase gevolgd door de implementatie van een Proof Of Concept (POC). Er zal gekeken worden naar de verschillende factoren die een rol spelen bij het implementeren van een dergelijk systeem. De keuzes die gemaakt worden tijdens de implementatie van het proof of concept zullen gebaseerd zijn op resultaten van het onderzoek.

Er is geen vooronderzoek gedaan door ALTEN op het gebied van locatiebepaling met geluidsdetectoren dat gebruikt kan worden voor het onderzoek. Er is geen werk verricht voordat er begonnen wordt aan het project. Al het onderzoekende en uitvoerende werk zal tijdens de loop van dit project gedaan worden. Er zijn meerdere onderzoeken te vinden waarin gedeelten van het project behandeld worden. Deze onderzoeken zullen gebruikt worden om de basis te leggen voor het literatuuronderzoek. Vanuit deze basis zal vervolgens meer onderzoek gedaan worden naar de andere aspecten van het project.

In het POC wordt een software gestuurde hardware architectuur gerealiseerd. Dit systeem zal bestaan uit losse geluidsdetectoren en minimaal 1 apparaat bedoelt voor het monitoren en besturen van de geluidsdetectoren. Uit onderzoek zal blijken welk aantal geluidsdetectoren het meest geschikt is, en wat voor type geluidsdetectoren het beste geschikt zijn.

Om het systeem met verschillende omgevingsvariabelen succesvol te kunnen laten werken, zal er een manier bedacht moeten worden om het systeem te kalibreren. Er zal initieel vooral handmatig gekalibreerd worden. Dit wordt gedaan door middel van het handmatig aanpassen van verschillende parameters binnen het systeem, en het plaatsen van de geluidsdetectoren op een voorbedachte plek. Welke parameters er precies gebruikt gaan worden, en welke er relevant zijn bij dit systeem zal blijken uit het onderzoek en testen van het POC. De nadruk ligt bij dit project op de software. Hier zal de nadruk vooral op liggen bij het implementeren van filtering, positiebepaling en de automatische kalibratie. Het creëren van een manier om automatisch te kunnen kalibreren zal waarschijnlijk het meest uitdagende gedeelte van het project zijn.

6.2 Afbakening

Hier wordt beschreven wat er wel en niet tijdens het afstudeerproject gedaan wordt.

6.2.1 Binnen scope

De onderstaande punten zijn onderdeel van wat er tijdens

- Onderzoek doen naar de verschillende aspecten van geluidsbron lokalisatie in 2D.
 - Omgeving.
 - Geluidseigenschappen.
 - Variabelen.
- Onderzoek naar geschikte hardware voor het POC.
 - Microfoons.
 - Kabels.
 - Centrale monitoring device.
- Onderzoek en ontwerp van geschikte software voor het POC.
 - Taal.
 - Structuur.
 - Conventies.
- Het realiseren van een POC.
 - Het koppelen van meerdere geluidsdetectoren aan een centrale monitoring device.
 - Het identificeren of een geluid dat gedetecteerd is relevant is voor de implementatie van het systeem. Wat de criteria is of een geluid relevant is of niet, zal bepaald worden met behulp van onderzoek. Denk hierbij bijvoorbeeld aan klappen.
 - Het bepalen van de locatie van een geluidsbron met het systeem in een afgesloten ruimte.
 - Het bepalen van de locatie van een geluidsbron met het systeem in een gecontroleerde situatie, in de buitenlucht.
 - Een manier om het systeem automatisch te kunnen laten kalibreren. Hoe dit mogelijk is en welke variabelen er automatisch gekalibreerd zouden kunnen worden, wordt in het onderzoek behandeld.
 - Het verkrijgen van relevante positionele data uit het systeem.

6.2.2 Buiten scope

De onderstaande punten, en andere toepassingen die niet genoemd zijn in de voorafgaande paragraaf, zullen niet behandeld worden in dit project.

- Onderzoek doen naar lokalisatie in 3D
- Het identificeren wat de oorzaak van een geluid is
- Het voorspellen van een pad die de geluidsbron mogelijk af gaat leggen.

7 Onderzoeksvragen

In dit hoofdstuk wordt het relevante onderzoek behandeld. Dit zal gebeuren aan de hand van onderzoeksvragen. Deze onderzoeksvragen worden eerst beantwoord met behulp van literatuuronderzoek. In [hoofdstuk 11 Resultaten](#) zullen de onderzoeksvragen nogmaals behandeld worden, aan de hand van de resultaten van de tests die gedaan gaan worden met het POC.

Er worden bij het beantwoorden van de deelvragen verschillende onderzoeksmethoden (Onderzoeksmethoden, 2020) gebruikt.

Met 'realtime' wordt bedoeld dat de gegevens verkregen in het systeem direct verwerkt worden, en dat er binnen 2 seconden een resultaat te zien is.

Een locatie zal gerepresenteerd worden als een punt met een bepaalde X en Y waarde ten opzichte van het gebied waar gemeten wordt. De geluidsdetectoren zullen ook een X en Y waarde hebben. De coördinaten representeren meters.

7.1 Hoofdvraag

De hoofdvraag luidt: **Hoe accuraat is het mogelijk om de realtime locatie van een bewegende geluidsbron te bepalen met behulp van geluidsdetectoren?**

De hoofdvraag is opgedeeld in deelvragen zodat ze individueel onderzocht kunnen worden.

De Hoofdvraag zal bij [hoofdstuk 12 conclusie](#) behandeld worden en zal met behulp van de resultaten van de deelvragen beantwoord worden.

7.2 Deelvragen

De deelvragen die hier behandeld worden zijn:

- 7.2.1 Hoeveel invloed heeft de omgeving waarin het systeem geplaatst is op de accuraatheid van de metingen?
- 7.2.2 Welke lokaliseringmethode is het meeste geschikt om te gebruiken binnen het project?
- 7.2.3 Wat voor verschillende variabelen hebben invloed op het bepalen van een locatie van een geluidsbron?
- 7.2.4 Hoe accuraat kan de locatie van een statische geluidsbron bepaald worden met behulp van geluidsdetectoren?
- 7.2.5 Is het mogelijk om met een automatische kalibratie, betere parameters te kunnen verkrijgen voor het bepalen van een locatie van een geluidsbron, dan met een handmatige kalibratie?

7.2.1 Deelvraag 1: Omgeving

Hoeveel invloed heeft de omgeving waarin het systeem geplaatst is op de accuraatheid van de metingen?

Aanpak

Eerst wordt er literatuuronderzoek gedaan naar de verschillende soorten invloeden van omgeving op geluid. Daarna gaat er getest worden in verschillende situaties. Met behulp van de resultaten van het onderzoek en de testen zal uiteindelijk een conclusie getrokken worden.

Resultaat

Geluidssnelheid

De geluidssnelheid wordt beïnvloed door de omgeving.

De temperatuur van de omgeving is een van de omgevingsfactoren die invloed heeft op geluidssnelheid. (Hannah, 2006) De geluidssnelheid in droge lucht is ongeveer:

$$v_{\text{sound in air}} \approx 331.4 + 0.6T_C \text{ m/s}$$

Figuur 1 Snelheid van geluid in droge lucht (hyperphysics, 2007)

Bij deze formule is T_C de temperatuur in graden Celsius ($^{\circ}\text{C}$). Er zal binnen de implementatie rekening gehouden worden met deze formule. Temperatuur is een omgevingsvariabele die lastig op verschillende manieren te testen is, omdat de temperatuur niet makkelijk te manipuleren is. Hierom zal er niet bij de praktijktesten over de temperatuur een goed onderbouwde conclusie getrokken kunnen worden.

De luchtvochtigheid heeft ook invloed op de snelheid van het geluid. Echter, luchtvochtigheid heeft een insignificant klein effect op de snelheid van het geluid. In Figuur 2 is te zien hoeveel invloed het heeft.

Temperature ($^{\circ}\text{C}$) (deg F)	Speed of Sound (m/s) (ft/min)								
	Relative Humidity (%)								
	10	20	30	40	50	60	70	80	90
0	331.5	331.5	331.5	331.6	331.6	331.6	331.7	331.7	331.7
5	334.5	334.6	334.6	334.7	334.7	334.7	334.8	334.8	334.9
10	337.5	337.6	337.7	337.7	337.8	337.9	337.9	338.0	338.0
15	340.5	340.6	340.7	340.8	340.9	341.0	341.1	341.2	341.2
20	343.5	343.6	343.7	343.9	344.0	344.1	344.2	344.4	344.5
25	346.4	346.6	346.8	347.0	347.1	347.3	347.5	347.6	347.8
30	349.4	349.6	349.9	350.1	350.3	350.5	350.8	351.0	351.2

Figuur 2 Invloed van luchtvochtigheid op de snelheid van het geluid. Figuur van (Engineering ToolBox, 2003)

Omdat de luchtvochtigheid zo weinig invloed heeft op de snelheid van het geluid, zal er geen rekening mee gehouden worden bij de implementatie van het POC.

Wind

Wind heeft invloed op geluid. (Hannah, 2006) De windsnelheid in de periode dat er buiten getest gaat worden (mei en juni) ligt in het gebied waar getest gaat worden gemiddeld tussen de 3 en 5m/s. (kaart gemiddelde windsnelheid, 2020)

Bij 20°C is de snelheid van het geluid ongeveer 343m/s. Aan de hand van de windrichting en de windsnelheid kan de snelheid van het geluid tot op de windsnelheid toenemen of

afnemen. Daardoor kan het verschil in aankomst tussen de geluidsdetectoren onvoorspelbaar fluctueren.

De richting waarop het geluid zich voortbeweegt kan ook veranderen. Hierdoor is het mogelijk dat een microfoon die zich verder van de geluidsbron af bevindt, het geluid toch eerder detecteert dan een microfoon die zich dichterbij de geluidsbron bevindt.

Weerkaatsing van geluid

Om extra inzicht te krijgen over de effecten van weerkaatsing en over geluid in het algemeen, is er om advies gevraagd bij de medewerkers van ALTEN. Hieruit zijn een aantal inzichten naar voren gekomen over weerkaatsing.

De weerkaatsing van geluid op hout is gemiddeld ongeveer 90%. (MBI products, 2001) Dit betekent dat 10% van het geluid geabsorbeerd wordt door het hout. Dit percentage verschilt op basis van de frequentie van het geluid. Dit is voor elk materiaal verschillend. Dit zou invloed kunnen hebben op het correct identificeren van het moment van waarnemen. Dit heeft vooral in een afgesloten ruimte een invloed, waar muren, het plafond en de grond allemaal kunnen zorgen voor extra weerkaatsing van geluid. Ook weerkaatsing op obstakels zou kunnen zorgen voor onvoorspelbare veranderingen in de detectie van geluid.

Wanneer er veel weerkaatsing in een ruimte voorkomt, is er ook een grotere kans op interferentie. Wanneer er sprake is van interferentie, kunnen de geluidsgolven overlappen en daardoor een andere frequentie vormen. Met perfecte interferentie kan de frequentie halveren of het geluid zelfs compleet dempen.

Fun fact: Interferentie is de manier waarop veel noise-canceling headphones geluid kunnen dempen. Door een omgekeerd geluid dan die wordt waargenomen af te spelen, lijkt het alsof het stil is.

Conclusie

De temperatuur is een belangrijke factor in het bepalen van de geluidssnelheid. De wind is een onvoorspelbare factor, maar kan wel degelijk een grote impact hebben op de manier waarop geluid zich voortbeweegt. De weerkaatsing van het geluid kan in galmende omgevingen erg veel invloed hebben op geluidsdetectie en identificatie. Al deze factoren in acht te hebben genomen, is de voorspelling dat het correct identificeren wanneer een geluid precies wordt waargenomen in een binnen-omgeving, een stuk moeizamer zal gaan dan in een buiten-omgeving.

7.2.2 Deelvraag 2: Lokaliseringsmethode

Welke lokaliseringsmethode is het meeste geschikt om te gebruiken binnen het project?

Aanpak

Om te beginnen er zijn een aantal methoden waarmee er in 2D een locatie bepaald kan worden. Elk van deze methoden heeft zijn voor en nadelen die hier behandeld worden. Voor 3D lokalisatie zijn er ook een aantal opties om te onderzoeken en implementeren. (3D_sound_localization - Wikipedia, 2020) Voor dit project zit dat buiten de scope, en zal het ook niet verder behandeld worden.

Resultaat

SRP-PHAT

SRP-PHAT (Steered Response Power Phase Transform) is een populair algoritme voor het lokaliseren van een geluidsbron. SRP-PHAT staat er om bekend robuust te zijn in onstuimige akoestische omgevingen. (Hoang Do, 2007) (Maximo Cobos, 2017)

Om SRP-PHAT te kunnen gebruiken, heb je de TDOA (Time difference of arrival) van microfoonparen nodig. Dat is het verschil in tijd tussen het waarnemen van geluidsgolven op 2 verschillende microfoons

Bij SRP-PHAT wordt er gekeken naar de mogelijkheid dat een geluidbron zich bevindt op een bepaald punt op een grid. Dit wordt gedaan door naar de TDOA's te kijken, en die te vergelijken met elk punt op een x en y grid. De TDOA's worden bepaald door een cross-correlation uit te voeren tussen elk paar van de ontvangen geluidssignalen. Cross-correlation is een algoritme die gebruikt wordt om geluidsoptnames met elkaar te vergelijken, en te berekenen op welk punt deze twee geluidsoptnames het meest met elkaar overeenkomen.

Er wordt RMSE gebruikt om een waarde te koppelen aan elk punt.

De RMSE wordt berekend door de berekende afstandsverschillen vanaf de microfoons, op basis van TDOA, te vergelijken met de afstandsverschillen bij elk x, y coördinaat. Eerst wordt de afstand bepaald van het punt naar elke microfoon (te zien in Figuur 3).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Figuur 3 Afstand formule, plaatje door (Chili math, 2011-2020)

Daarna worden deze afstanden van de berekende afstanden afgehaald en wordt de RMSE berekend met de formule te zien in Figuur 4.

$$rmse = \sqrt{(d_{01}^2 + d_{02}^2 + d_{12}^2) / 3}$$

Figuur 4 RMSE Bij SRP-PHAT

Waarbij d staat voor het absolute afstandsverschil tussen, de microfoons met de aangegeven microfoon nummers in subscript, en het berekende afstandsverschil met TDOA.

SRP-PHAT is waarschijnlijk het beste geschikt voor deze applicatie aangezien het robuuster en betrouwbaarder is dan de meeste andere locatiebepalingsmethoden die mogelijk zijn in deze setting. SRP-PHAT is wel een resource intensieve methode, wat impact heeft op de snelheid van de meting wanneer er een locatie preciezer bepaald wordt.

Triangulatie

Triangulatie, ook wel Driehoeks bepaling, is een manier om de locatie van een punt te kunnen berekenen met behulp van een driehoek die een geluidsbron als laatste punt van de driehoek beschouwt. Als je de hoek van inval weet vanaf twee verschillende punten, en je weet de afstand tussen de punten, is het mogelijk het punt te berekenen van de geluidsbron. Dit is niet gelimiteerd tot geluidssensoren en een geluidsbron, maar ook mogelijk met afstandmeters, camera's en andere soort sensoren.

Trilateration

Voor Trilateration is de tijd van creatie van het geluid nodig. (Dalskov, 2014) Voor de toepassing die nodig is voor dit project is dat niet mogelijk. Trilateration zal dus niet gebruikt gaan worden.

Multilateration

Bij multilateration kan er een locatie berekend worden op basis van 2 microfoon paren. Hierbij kan van deze paren 1 microfoon dezelfde zijn. Op basis van de TDOA's tussen de verschillende microfoonparen, kunnen er hyperbolen gedefinieerd worden die op elk punt dezelfde TDOA's hebben in vergelijking tot de microfoons. De locatie van het geluid kan berekend worden door deze hyperbolen van beide paren te vergelijken en de kruising van de 2 te bepalen. (Dalskov, 2014)

Binaural Recording

Binaural Recording kan beschreven worden als het nabootsen van het hebben van 2 oren. Hiermee kunnen hoeken en plaatsen van geluidsbronnen bepaald worden. (Paul, 2009) Voor dit project is dit niet een methode die interessant is om verder te onderzoeken, omdat er een bepaalde nauwkeurigheid bereikt wil worden, die met Binaural Recording niet te bereiken is.

Conclusie

Om de redenen genoemd bij de verschillende methoden is er uiteindelijk gekozen om het SRP-PHAT algoritme te gaan gebruiken als lokaliseringmethode. Omdat er bij de implementatie is gekozen om een keuze te kunnen maken tussen meerdere lokaliseringmethoden (hot-swappable), is er voor gekozen om triangulatie als een tweede lokaliseringmethode te implementeren.

7.2.3 Deelvraag 3: Aanpasbare variabelen

Wat voor verschillende variabelen hebben invloed op het bepalen van een locatie van een geluidsbron?

Aanpak

Er zal deskresearch gedaan worden om te kijken welke parameters en variabelen mogelijk een rol spelen bij het bepalen van de locatie van een geluidsbron. Tijdens de implementatie van het POC zullen, naast wat er uit deskresearch komt, mogelijk nog een aantal variabelen ter tafel komen die belangrijk zijn voor het bepalen van een locatie van een geluidsbron.

Resultaat

Geluidssnelheid

Zoals genoemd bij [deelvraag 1](#), heeft de omgeving invloed op de geluidssnelheid. Temperatuur is relatief makkelijk te meten en heeft een significante invloed op de geluidssnelheid. Om deze redenen gaat temperatuur gebruikt worden als enige parameter van de geluidssnelheid bij het POC.

Gain

Een parameter die in de context van deze setup aangepast kan worden, is de gain. Het aanpassen van gain is een hardware aspect, in tegenstelling tot een parameter die in te stellen is in de software. Hierdoor kan dit niet ingesteld worden bij een automatische kalibratie.

Afstand tussen microfoons

Wanneer er geprobeerd wordt om een locatie te bepalen op basis van TDOA's, moet de afstand tussen de microfoons bekend zijn. Wanneer de afstand bekend is kunnen TDOA's gebruikt worden om afstandsverschillen te vergelijken met de afstanden tussen de microfoons.

Lokaliseringsmethode

Bij het gebruiken van verschillende lokaliseringmethoden zijn er verschillende parameters waar rekening mee gehouden moet worden. Bij SRP-PHAT is het bijvoorbeeld belangrijk tot op welk bereik je punten wilt gaan berekenen. Bij Trilateration is het bijvoorbeeld nodig om te weten wat de tijd van creatie van het geluid is.

Conclusie

De parameters die belangrijk zijn voor het bepalen van een geluidsbron, en die realistisch gezien gebruikt kunnen worden in het POC, zijn geluidssnelheid en de afstand tussen de microfoons.

7.2.4 Deelvraag 4: Precisie

Hoe accuraat kan de locatie van een statische geluidsbron bepaald worden met behulp van geluidsdetectoren?

Aanpak

Met deskresearch zal gekeken worden welke factoren er theoretisch invloed zouden moeten hebben op de precisie van lokalisering.

Resultaat

Hoogte van de geluidsbron

Wanneer er een geluid gemaakt wordt op een andere hoogte dan de microfoons, kan het zijn dat de resultaten niet helemaal overeenkomen met een 2D locatie. Bijvoorbeeld op een hoogteverschil van 2 meter tussen het geluid en de microfoons. Als de afstand tot microfoon 1 in 2D 5 meter is, zal dat in 3D ongeveer 5.385 meter zijn. Als de afstand tot microfoon 2 in 2D 3 meter is, zal dat in 3D ongeveer 3.6 meter zijn. Het verschil tussen de afstanden is dan significant kleiner dan wanneer je dat in 2D zou bekijken. Hier zal rekening mee gehouden moeten worden in het testplan, en er zal gekeken worden hoe hiermee rekening gehouden kan worden bij de berekeningen van de methoden. Wanneer er een vierde microfoon gebruikt wordt en ook in de hoogte voorspeld wordt, is het mogelijk om een 3D locatie te kunnen bepalen.

Frequentie van geluiden

Sommige geluiden hebben een bepaalde frequentie range waaraan te zien is wat voor geluid het mogelijk kan zijn. Hiermee kan dus ook bepaald worden of een geluid interessant is. Aan het begin gaan klappen en vingers knippen aangehouden worden voor het detecteren van een interessant geluid. De frequentie range van in de handen klappen ligt gemiddeld tussen de 2200Hz en 2800Hz. (Hopper, 2010).

De frequentie range van het knippen in de vingers ligt tussen de 1500Hz en de 3500Hz. (LOKKI, 2010)

Afstand van het geluid

De afstand tussen het geluid en de microfoons kan ook invloed hebben op hoe accuraat de meting is. Vooral wanneer het geluid niet door alle microfoons even duidelijk gedetecteerd wordt. Wanneer dit gebeurt kan de tijd van aankomst op elke microfoon niet accuraat meer bepaald worden.

Sample rate

Bij het nemen van audio samples, is het belangrijk om te weten hoe lang het heeft geduurd om die audio samples te nemen. Wanneer het aantal samples gedeeld wordt door de tijd dat het gekost heeft om deze samples te verkrijgen, komt er een sample rate uit. Wanneer er een lage sample rate is, kunnen de hoge frequenties niet goed meer geïdentificeerd worden. Een frequentie kan geïdentificeerd worden wanneer de sample rate ten minste twee keer zo hoog is dan de frequentie. Dus wanneer de frequentie 5.000Hz is, moet de sample rate ten minste 10.000 samples/s zijn.

Naast dat het voor het detecteren van een frequentie nodig is om een bepaalde sample rate te hebben, is het ook nodig voor het accuraat meten van het TDOA tussen de microfoons.

Voorbeeld: De sample rate is 10.000 samples/s en de geluidssnelheid is 343m/s. Dat betekent dat elke keer dat 1 sample gemeten wordt, er 3.43cm afgelegd wordt door het geluid. Wanneer er een afstand tussen twee microfoons zit van 10cm, zitten er dus maximaal maar 3 samples verschil tussen de metingen. Dat betekent dat er maar 3 mogelijke hoeken van inval zijn die tussen de microfoons bepaald kunnen worden. Wanneer de sample rate 100.000 samples/s is, betekent dat maximaal 30 samples verschil en 30 mogelijke hoeken van inval zijn. Hiermee kan veel accurater een locatie bepaald worden.

Afstand tussen de microfoons

Zoals genoemd bij het voorbeeld van de sample rate, heeft de afstand tussen microfoons ook invloed op de precisie van een meting. Bij een kleinere afstand tussen de microfoons, zullen er ook minder en kleinere TDOA's mogelijk zijn.

Maar bij grotere afstanden tussen de microfoons is er een grotere kans dat niet elke microfoon het gemaakte geluid detecteert. Ook moeten er meer samples gemeten worden om het te detecteren geluid ontvangen te hebben op alle microfoons bij een grotere afstand.

Omgeving

De omgeving waarin een meting plaats vindt heeft invloed op de precisie van de meting. Kijk naar [Deelvraag 1](#) voor meer informatie over de invloed van de omgeving.

Lokaliseringsmethode

De gebruikte lokaliseringmethode heeft invloed op de precisie van een meting. Kijk naar [Deelvraag 2](#) voor meer informatie over lokaliseringmethoden.

Conclusie

De precisie bij het bepalen van de locatie van een geluidsbron is afhankelijk van een overvloed aan factoren die invloed hebben op de meting die gedaan wordt. Er wordt verwacht dat de omgeving, in combinatie met de afstand tussen microfoons en de lokaliseringmethode samen de grootste invloed hebben op de precisie van de locatiebepaling.

7.2.5 Deelvraag 5: Kalibratie

Is het mogelijk om met een automatische kalibratie, betere parameters te kunnen verkrijgen voor het bepalen van een locatie van een geluidsbron, dan met een handmatige kalibratie?

Aanpak

Om te kijken of het mogelijk is om een automatische kalibratie te realiseren, zal er tijdens de implementatie van het POC, geprobeerd worden een kalibratie methode te ontwikkelen. Om te controleren of het werkte, en hoe goed het werkt, zullen er praktijktests uitgevoerd worden met het POC.

Resultaten

Variabelen

De variabelen die, zonder de hardware direct aan te passen, ingevoerd kunnen worden door de gebruiker zijn besproken in [Deelvraag 3](#). Dit zijn de temperatuur, de lokaliseringmethode en de afstand tussen de microfoons.

Handmatige kalibratie

Bij handmatige kalibratie zullen de variabelen die nodig zijn om lokaliseringen te kunnen doen, ingevoerd worden door de gebruiker van het systeem. Hiervoor moet de afstand tussen de microfoons dus met een meetlint of iets dergelijks gemeten en ingevoerd worden.

Automatische kalibratie

Wanneer er een thermometer verbonden zou zijn aan het systeem, zou de geluidssnelheid automatisch bepaald kunnen worden. Omdat de focus niet op hardware ligt in dit project, is er voor gekozen om temperatuur niet als automatisch kalibreerbare parameter te gebruiken. Bij de automatische kalibratie moet de temperatuur dus ingevoerd worden door de gebruiker.

Welke lokaliseringmethode er gebruikt gaat worden zal ook door de gebruiker gekozen worden. Hierbij krijgt de gebruiker opties om uit te kiezen.

De informatie die bij de automatische kalibratie bepaald wordt, zijn de afstanden tussen de microfoons.

Automatische kalibratie stap voor stap

1. Voer de temperatuur in
 - (a) De temperatuur wordt gebruikt om de geluidssnelheid te berekenen
2. Klap 30cm boven elke microfoon.
 - (a) De frequentie van het geluid wordt bepaald bij elke microfoon voor elke klap.
 - (b) De TDOA's tussen de microfoons worden berekend voor elke klap.
 - (c) De frequenties worden met elkaar vergeleken om te bepalen of het geluid binnen een marge correct genoeg gedetecteerd is op elke microfoon.
 - (d) Als er aan die voorwaarde is voldaan, wordt de microfoon waarbij het geluid gemeten is als klaar beschouwt
3. Wanneer dit voor alle 3 de microfoons gedaan is
 - (a) Worden de verschillende TDOA's met elkaar vergeleken en wordt er een gemiddelde afstand genomen die als afstand tussen het microfoonpaar gebruikt wordt
 - (b) Als de TDOA's tussen dezelfde microfoons bij verschillende metingen te ver uit elkaar liggen, wordt de kalibratie als gefaald beschouwd. Als dit zo is moet er weer begonnen worden bij stap 2.
4. De applicatie berekend de coördinaten van de microfoons.
5. De gebruiker kiest een lokaliseringmethode die gebruikt moet worden.
6. De automatische kalibratie is nu voltooid, vanaf dit punt wordt voor geluid gesampled.

Wanneer het systeem gebruikt gaat worden met automatische kalibratie, zal de kalibratie elke keer opnieuw uitgevoerd moeten worden wanneer de windsnelheid of windrichting veranderd. Het is lastig om rekening te houden met de wind omdat het erg te maken heeft met de omgeving waar het systeem zich in bevindt, en deze erg kan fluctueren. Ook is het lastig is om een windrichting en windsnelheid op de precieze plek van de metingen te kunnen bepalen.

Conclusie

Bij automatische kalibratie kan de afstand tussen de microfoons tot op zekere hoogte worden bepaald. Hierbij hoeft de gebruiker alleen een aantal non-complexe stappen door te lopen zodat het systeem, aan de hand van de resultaten die daaruit komen, de waarden kan bepalen. Realistisch gezien worden deze waarden niet zo accuraat als bij het meten van de afstand met een meetlint. De hoop is dat de automatische kalibratie dichtbij genoeg bij de echte waarden kan komen dat het geen significante impact zal hebben op de meetresultaten.

8 Hardware

Om praktijkgericht verder op de onderzoeksvragen in te kunnen gaan, moet er een idee zijn van welke componenten er gebruikt gaan worden en hoe ze gebruikt gaan worden. In dit hoofdstuk worden de belangrijkste hardware aspecten behandeld. Hierin wordt onderzocht wat de beste opties zijn voor de microfoons, de kabels en de Central monitoring device.

8.1 Microfoons

Aanpak

Met behulp van deskresearch zal er onderzocht worden hoeveel geluidsdetectoren er theoretisch gezien nodig zijn om een locatie van een geluidsbron te kunnen bepalen. Vervolgens wordt er onderzocht welke microfoons het meest geschikt zijn, en uiteindelijk zal er als conclusie een keuze worden gemaakt worden welke microfoons gebruikt gaan worden bij het POC

Resultaat

Eerst bestond het idee om usb microfoons te gebruiken. Na onderzoek bleek dat het protocol voor usb te langzaam data verstuurd om een timing-critical systeem te kunnen verwezenlijken. (Raphael Wimmer, 2019)

Hoeveelheid

Een 1D locatie (dat de locatie van de geluidsbron op een bepaalde lijn zit) kan bepaald worden met 2 microfoons (Maximo Cobos, 2017).

Een 2D locatie kan bepaald worden met 3 microfoons (Maximo Cobos, 2017)

Een 3D locatie kan bepaald worden met 4 Microfoons (3D_sound_localization - Wikipedia, 2020).

Analoog vs digitaal signaal

Bij Analoge signalen wordt de data in de vorm van veranderingen in voltage verstuurd. Digitale signalen worden verstuurd in de vorm van data met enen en nullen. Dit gebeurt met een aantal hele scherpe pieken in voltage die bij aankomst omgezet worden in data. (BJC, 2017)

Analoge signalen hebben een grotere precisie op timing, omdat digitale signalen eerst omgezet moeten worden naar analoge signalen. Hier komt bij dat er een grotere kans is op ruis en verlies van signaal.

Digitaal is erg betrouwbaar. Op het moment dat er zich ruis voordoet zal het digitale signaal nog steeds goed geïnterpreteerd kunnen worden.

Omdat het bij dit project vooral gaat om de timing en het verschil in aankomst, zullen er analoge Microfoons gebruikt worden.

MEMS vs ECM

Micro Electro Mechanical System (MEMS) en Electret Condenser Microphones (ECM) zijn twee soorten microfoons die vaak gebruikt worden in projecten die simpele microfoons nodig

hebben. (Rose, 2020) MEMS' zijn over het algemeen nieuwer en beter, maar er zijn ook toepassingen waarin een ECM een betere optie kan zijn. ECM's zijn over het algemeen goedkoper, en er zijn meer varianten te koop. Het gaat er bij het POC niet om de optimale setup te creëren, en daarom is er gekozen om ECM's te gebruiken tijdens dit project.

Gain

Er moet rekening gehouden worden met de gain van microfoons. (Eargle, 2004) De gain van een microfoon heeft invloed op hoe zacht het geluid kan zijn als de microfoon hem nog kan detecteren. Meer gain betekent dat de microfoon zachtere geluiden kan detecteren. Wanneer de gain ingesteld wordt, zal het ook mogelijk zijn om geluid dat van verder weg komt te detecteren. Om deze redenen is er gekozen om microfoons met instelbare gain te gebruiken.

Conclusie

Er is uiteindelijk gekozen voor een ECM op een amp breakout met instelbare gain (Maxim Integrated Products, Inc, 2012) (CUI Devices, 2020). Hiervan worden er 3 gebruikt met back-up van een vierde microfoon.

8.2 Kabels

Het type kabel dat gebruikt gaat worden, worden uitgezocht op kwaliteit en beschikbaarheid. Hierbij zijn een aantal aspecten die invloed hebben op de kwaliteit van het type kabel.

Noise reduction

Er zijn verschillende manieren om met noise om te gaan of noise tegen te gaan. Om ground noise te voorkomen kan er een losse ground kabel aangesloten worden voor de stroomtoevoer. Dit kan wanneer er minimaal 3 draden in de kabels aanwezig zijn. Om ruis over de kabel te voorkomen kan er gebruikt gemaakt worden van shielding. Hiermee zit er materiaal om de draden binnen de kabel heen om er voor te zorgen dat er zo weinig mogelijk kan gebeuren met het signaal dat over de draden gaat. Om omgevingsruis op de kabel te voorkomen kan er een gebalanceerde kabel en microfoon gebruikt worden. (Aviom, 2015) Een gebalanceerde kabel heeft 2 draden voor de data. 1 met het signaal en 1 met het signaal omgekeerd. Hierdoor kunnen die signalen met elkaar vergeleken worden en kan de ruis eruit gehaald worden. Om gebalanceerde microfoons te kunnen gaan gebruiken moet er werk verricht worden om dit te verwezenlijken met een microcontroller en 3 microfoons. Om te zorgen dat de focus niet te veel op hardware komt te liggen, is er voor gekozen om geen microfoons te gebruiken die een gebalanceerd signaal produceren.

Voltage drop

Bij elke kabel is er een vorm van voltage drop. Dit komt door de elektrische weerstand van draden. Bij de langere kabels moet hierom gedacht worden aan kabeldikte en materiaal. Een dikkere kabel zorgt voor minder voltage drop en zorgt dus ook voor een sterker signaal. (voltage drop calculator, 2020). Het is belangrijk dat het signaal zo veel mogelijk overeenkomt als met het geluid wanneer het is waargenomen, want dat maakt het makkelijker om de geluidsopnames met elkaar te vergelijken.

Conclusie

Er zijn een hoop soorten kabels. Het idee is om hiervoor kabels te gebruiken die al gebruikt worden voor geluid en microfoons. (Daddy, 2016) De best geschikte kabels zijn XLR kabels. Ze hebben veel ingebouwde shielding, zijn beschikbaar in grote lengtes en hebben 3 draden die gebruikt kunnen worden. De connector zelf is niet per se nodig, maar kan gebruikt worden om het kabelsysteem “plug and play” te maken. De draden die in een XLR kabel zitten worden standaard gebruikt om een gebalanceerd signaal te creëren, maar kunnen ook als VCC, GND en OUT te functioneren.

8.3 Central monitoring device

Omdat er meerdere microfoons verbonden moeten worden aan 1 systeem, is het nodig om een device te plaatsen die zorgt voor het verzamelen van de audio data. Dit device moet data kunnen versturen naar een laptop, timing critical data collection kunnen uitvoeren en moet ADC's hebben met een sample rate die hoog genoeg, zoals behandeld in [deelvraag 4](#), kan worden.

Esp, Intel quark, Pine64, Asus tinkerboard

Deze bordjes zouden allemaal geschikt kunnen zijn voor deze toepassing. Voor deze bordjes zijn weinig of geen voorbeeldprojecten die met het processen van meerdere geluidssignalen bezig zijn geweest. Gezien de complexiteit van het project, en de moeite die genomen zou moeten worden om de bordjes te kunnen gebruiken, worden deze niet gebruikt.

Raspberry Pi

Het grootste probleem bij een Raspberry Pi is dat ze standaard geen ADC's erop hebben zitten. Dit zorgt er voor dat er een digitale microfoon gebruikt moet worden, of dat er ADC's los bij gebruikt moeten worden. Aangezien er gekozen is om het zo simpel mogelijk te houden en analoge microfoons te gebruiken, gaat er niet gebruik gemaakt worden van een Raspberry pi.

Arduino

Arduino zou een geschikte kandidaat kunnen zijn voor deze toepassing. De Arduino heeft veel community support, meerdere libraries die gebruikt kunnen worden van het processen van geluid en een groot aantal voorbeeldprojecten die vergelijkbare toepassingen hebben als wat er met dit project bereikt moet worden. Ook zitten er genoeg aansluitingen op om meerdere microfoons op aan te sluiten.

Teensy

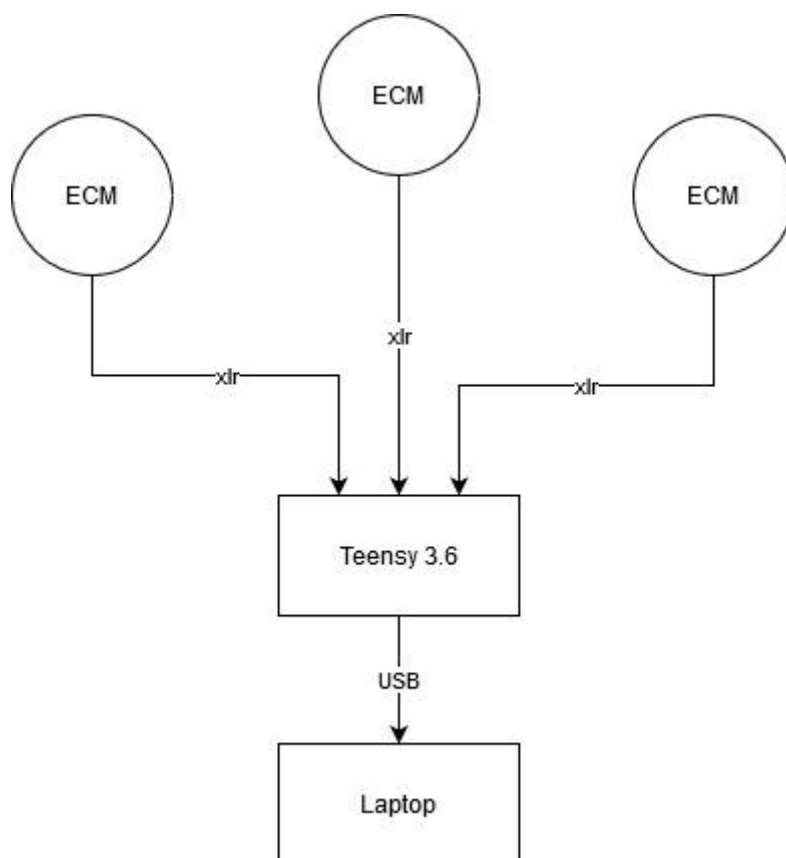
De Teensy heeft alle voordelen van een Arduino, maar dan met betere specs. (prjc, 2020) Er is een Teensy plug-in ondersteund door de maker van de Teensy, waarmee Arduino sketches ook op de Teensy gezet kunnen worden. De Teensy kan een hoge sample rate halen met de ADC's die gebruikt worden, wat erg belangrijk is bij timing critical toepassingen. Ook hebben de meeste Teensy boards voldoende aansluitingen voor meerdere microfoons. Dit maakt de Teensy uitermate geschikt voor dit project.

Conclusie

Door de hoge sample rate, alle benodigde microfoons aan te kunnen sluiten, en de mogelijkheid van het gebruiken van Arduino sketches, is er gekozen voor een Teensy 3.6 als central monitoring device.

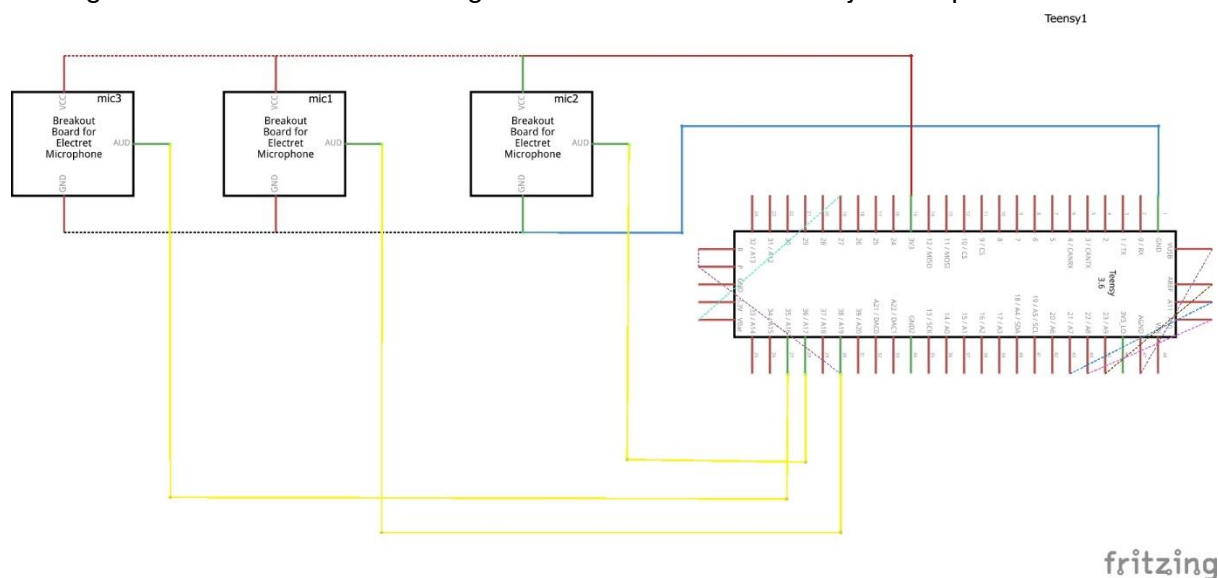
8.4 Hardware design

Hier wordt een globaal overzicht van hoe de hardware setup eruit gaat zien gegeven. In Figuur 5 is te zien dat de ECM's met XLR kabels verbonden zijn met de Teensy, en de Teensy is vervolgens met de laptop verbonden met een USB kabel.



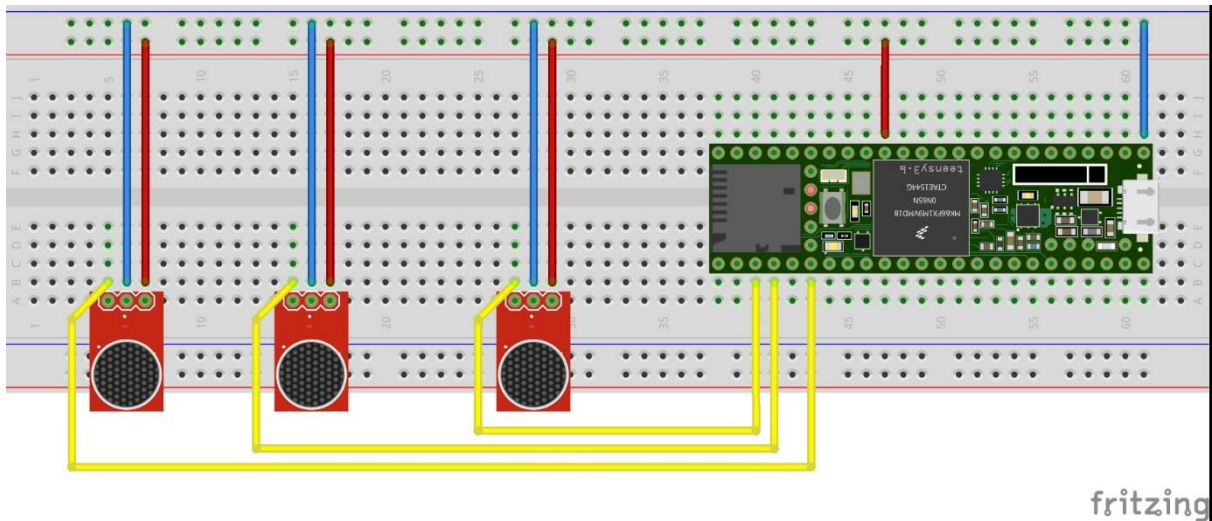
Figuur 5 Abstract overzicht hardware componenten

In Figuur 6 is het interne bedradingsschema te zien die hoort bij deze opzet.



Figuur 6 Interne bedradingsschema

In Figuur 7 is te zien hoe het breadboard bij de eerste setup in elkaar zit. Blauwe draden zijn verbonden met de ground pin. Rode draden zijn verbonden met de 3.3V pin. Gele draden zijn verbonden met de analoge input pins.



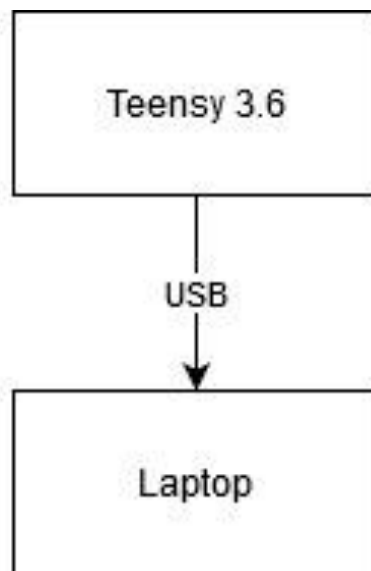
Figuur 7 Breadboard lay-out eerste opzet

9 Software

Nu er bepaald is wat voor hardware er gebruikt gaat worden, is het tijd om te kijken welke software hier het meest geschikt voor zou zijn. Het primaire doel van de software is een geluid kunnen detecteren en een inschatting van de locatie van de geluidsbron bepalen. Om dit te kunnen verwezenlijken moet de software gekalibreerd kunnen worden. Omdat het een proof of concept omtrent en niet een product, is er maar 1 soort gebruiker. Dit is de persoon die tests uitvoert en al voorkennis heeft van het systeem.

9.1 Architectuur

De software architectuur van dit project bestaat uit twee hoofdcomponenten. De software die draait op de Teensy en de desktop applicatie. Deze twee zijn verbonden via een seriële verbinding. Een seriële verbinding is een sequentiële verbinding die 1 bit tegelijkertijd 1 kant op kan versturen. Het is niet mogelijk om tegelijkertijd iets te verzenden en te ontvangen. In plaats hiervan wacht het ontvangende apparaat met het verzenden van zijn bericht tot hij de data uitgelezen heeft die ontvangen wordt. Aan het eind van elk karakter die verzonden wordt via deze seriële verbinding, wordt een stop bit meegestuurd. Deze stop bit is er voor bedoeld om het einde van een karakter te kunnen identificeren.



Figuur 8 Abstracte architectuur

9.2 Teensy

9.2.1 Requirements

ID	Beschrijving	Priority	Voldaan?
RT1	De Teensy moet geluidsdata analoog kunnen uitlezen vanaf de microfoons en tijdelijk kunnen opslaan	Must	Ja
RT2	De Teensy moet een sample rate hebben van minimaal 100.000 samples per seconde	Must	Ja
RT3	De Teensy moet de geluidsdata kunnen versturen naar de desktop applicatie via een seriële verbinding	Must	Ja
RT4	De Teensy moet commando's kunnen ontvangen van de desktop applicatie via een seriële verbinding	Should	Ja

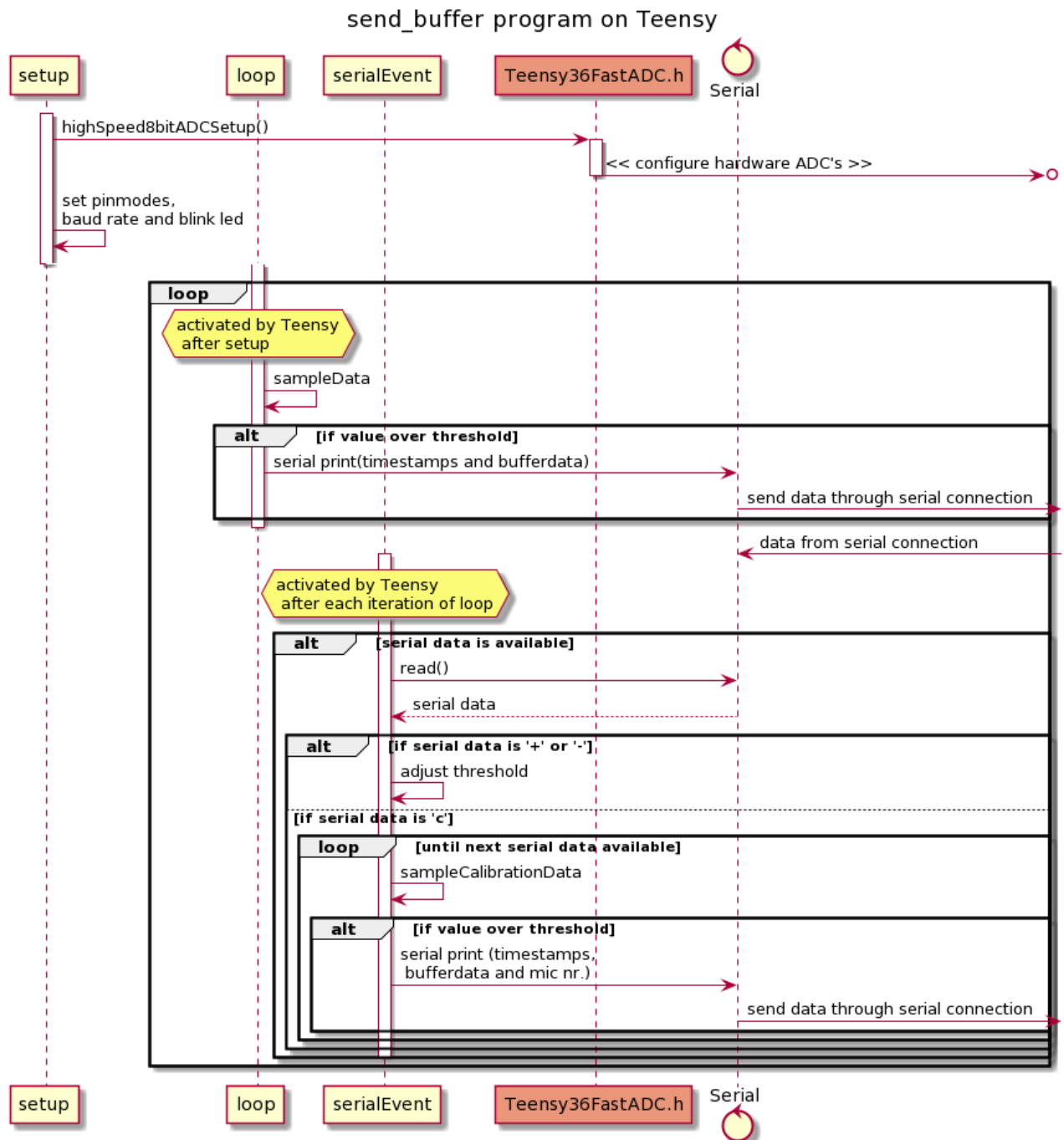
Tabel 3 - Software Requirements Teensy

9.2.2 Taal

De keuze voor de taal die gebruikt gaat worden op de Teensy is snel gemaakt. De standaard talen die op de Teensy ondersteund worden zijn c en c++. Mijn ervaring in deze talen is erg beperkt en zou de keuze op een van deze talen vallen, zou dat hun eigen set met obstakels met zich meebrengen. Zoals in [8.1.3, bij het kopie Teensy](#) genoemd, heeft de Teensy een plug-in die er voor zorgt dat Arduino sketches ondersteund worden. Dit maakt het een stuk gemakkelijker om code voor op de Teensy te schrijven. Omdat de Arduino community ongelooflijk groot is, helpt dat ook bij het vinden van oplossingen bij problemen waar mogelijk tegenaan gelopen wordt. Om deze redenen is er gekozen om de Teensyarduino plug-in te gebruiken om in Arduino sketches de software te kunnen schrijven.

9.2.3 Overzicht

De Software van de Teensy bestaat uit 3 onderdelen: Setup, loop en serialEvent.



Setup

Dit is waar de Teensy altijd begint wanneer de applicatie opstart. Bij de setup van de Teensy de LED pinMode naar output gezet, worden de input pins gezet, wordt de baud-rate van de seriële verbinding naar 2 miljoen gezet en wordt het LEDlampje 1 keer voor 2 seconden laten knipperen, om zo aan te geven dat de setup voltooid is. De baud-rate is gezet op de hoogst mogelijke baud-rate die de Teensy zonder error kan gebruiken (prjc, 2020).

Wat er nog meer gebeurt in de setup, is de `highSpeed8bitADCSetup`. Dit is een functie die geïmporteerd wordt uit de headerfile `Teensy36FastADC.h`. Deze headerfile heb ik niet zelf geschreven, en was bedoeld voor de Teensy 3.1. Deze heb ik geïmporteerd van een project op GitHub, en vervolgens aangepast zodat het zou werken op de Teensy 3.6. (Kriphor, `kriphor/teensy31-micloc Teensy 3.1 MicLoc project`, 2014) Met deze functie worden een aantal instellingen van de ADC's op de Teensy aangepast om een hogere sample rate te krijgen. De grootste aanpassing aan de ADC's is dat er hiermee in plaats van 32 bits integers, 8 bits integers gebruikt worden. Met behulp van deze veranderingen kunnen er 3 microfoons tegelijk gemiddeld tussen de 200.000 en 300.000 keer per seconden uitgelezen worden.

Loop

Loop wordt altijd uitgevoerd wanneer de setup klaar is. De loop herhaalt zichzelf eendeloos totdat de applicatie afgesloten wordt, of er een event tussendoor komt die er voor zorgt dat de loop stopt.

In het loop gedeelte zit een for-loop die de input van de microfoons uitleest tot de buffers van de respectievelijke microfoons gevuld zijn. Als er een waarde gedetecteerd wordt die boven een bepaalde threshold zit, initieert een event. Wanneer de buffers gevuld zijn, en er is een event geweest, zullen ze over de seriële poort verzonden worden. Dit gebeurt door voor elke waarde in een loop uit de buffer gelezen, geconverteerd naar een hexadecimaal getal, en over de seriële verbinding gestuurd. De plek waar de waarde net vandaan gelezen is wordt leeggemaakt en er wordt een spatie meegestuurd. Wanneer dit klaar is voor alle buffers, worden de start en stop time meegestuurd. Deze worden meegestuurd om te kunnen berekenen wat de loopduur van de geluidsopname was, en daarmee wat de sample rate was tijdens het opnemen. Dit gebeurt bij elke geluidsopname omdat de sample rate kan schommelen tussen metingen.

Bij het versturen van de buffers is er gekozen voor het gebruiken van hexadecimale waarden omdat deze minder ruimte in beslag nemen, en consistent zijn dan octale getallen.

serialEvent

De functie `serialEvent` wordt aangeroepen wanneer er data over de seriële port naar de Teensy verstuurd wordt. Wanneer dit gebeurt wordt er gekeken welk karakter ontvangen wordt.

Bij een '+' wordt de threshold waarop een event gedetecteerd wordt in de loop met 5 opgehoogd.

Bij een '-' wordt deze threshold met 5 verlaagd.

Wanneer de threshold aangepast is, wordt er een bericht met de nieuwe threshold over seriële verbinding teruggestuurd.

Wanneer er een 'c' ontvangen wordt, betekent dat dat de automatische kalibratie ingezet wordt. Wanneer dit gebeurt zal er vergelijkbaar gesampled worden als er in de loop gesampled wordt. Bij de kalibratie worden de samples, wanneer het event later dan de helft van de samples gedetecteerd is, verdubbeld. Maar bij de kalibratie wordt de microfoon waar het event gedetecteerd is ook meegestuurd, en wordt de grootte van de buffer verdubbeld in plaats van met de helft vergroot.

9.3 Desktop applicatie

9.3.1 Requirements

ID	Beschrijving	Priority	Voldaan?
RD1	De desktop applicatie moet seriële data kunnen ontvangen en verwerken	Must	Ja
RD2	De desktop applicatie moet de geluidsdata kunnen verwerken en vergelijken	Must	Ja
RD3	De desktop applicatie moet de geluidsdata kunnen verwerken en vergelijken. Hiermee moet ten minste de frequenties en de TDOA's van het geluid bepaald kunnen worden	Must	Ja
RD4	De desktop applicatie moet een inschatting geven van de locatie waar het geluid heeft plaatsgevonden gebaseerd op de verwerking van de geluidsdata, gebruik makend van de optie(s)	Must	Ja
RD5	De desktop applicatie moet met een handmatige kalibratie methode ingesteld kunnen worden	Must	Ja
RD6	De desktop applicatie moet met een automatische kalibratie methode ingesteld kunnen worden	Should	Ja
RD7	De desktop applicatie moet terwijl het data aan het ontvangen, is commando's van de gebruiker kunnen verwerken, en deze als nodig versturen naar de Teensy via een seriële verbinding	Should	Ja
RD8	De desktop applicatie moet data kunnen opslaan in een leesbaar format	Should	Ja
RD9	De desktop applicatie heeft een optie om tussendoor opnieuw te kunnen kalibreren, zonder het programma opnieuw op te starten	Could	Ja
RD10	De desktop applicatie heeft de optie om van lokaliseringmethode te wisselen zonder het programma opnieuw op te starten	Could	Ja

Tabel 4 Software Requirements Desktop applicatie

9.3.2 Talen

Wanneer er gekeken werd naar de mogelijkheden qua talen voor de software die op de laptop draait, zijn er eigenlijk maar twee talen in overweging genomen. Java en Python. De reden waarom er niet meer talen zijn overwogen is omdat mijn ervaring in deze talen significant genoeg is om een groot project mee te beginnen en deze talen beide geschikt kunnen zijn voor de software die geschreven moet worden.

De meeste van mijn ervaring zit in Javascript, maar dat is niet een geweldig geschikte taal voor het processen van grote hoeveelheden data.

Om uiteindelijk tussen Python en Java te kiezen, zijn er een aantal aspecten van de talen met elkaar vergeleken.

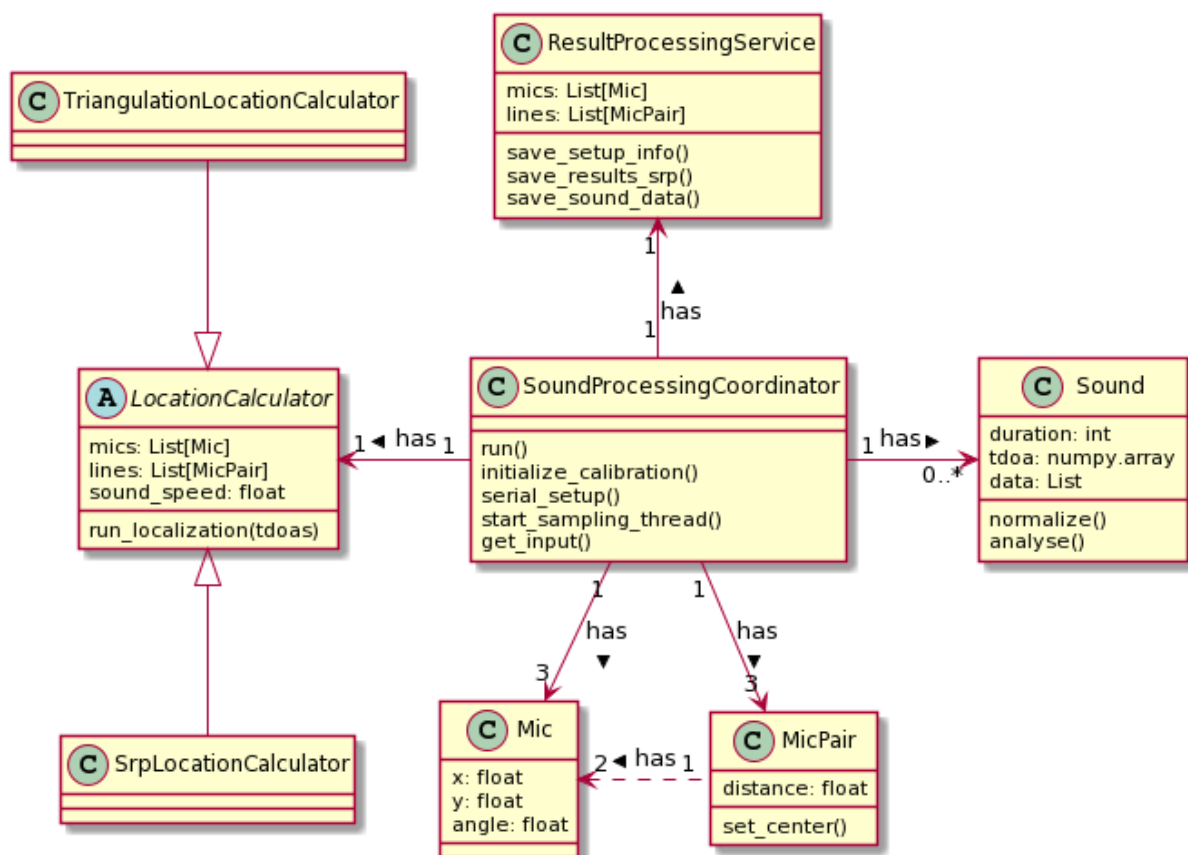
	Python	Java	Winnaar
Community	Stack Overflow vragen in maart 2020: ~15% van totaal	Stack Overflow vragen in maart 2020: ~7% van total	Python

	GitHub pull requests kwartaal 1 van 2020: ~16% van totaal	GitHub pull requests kwartaal 1 van 2020: ~11% van totaal	
Code	Compact, minder lines code voor dezelfde functionaliteit. Overzichtelijk door indentatie gebaseerde syntax.	Verbose syntax, veel code voor de functionaliteit die er aan verbonden is. Hierdoor kan andermans en/of legacy code lastiger te begrijpen zijn.	Python
Snelheid	Omdat het een interpreted language is het runtime minder snel dan Java.	Omdat het een compiled language is, duurt het compilen langer, maar is het runtime sneller dan python	Java
Typing	Maakt gebruik van dynamic typing	Maakt gebruik van static typing	-

Tabel 5 Python vs Java. Data: (Zapponi, 2020) (stackoverflow insights, 2020)

Uiteindelijk is er gekozen om Python te gebruiken als programmeertaal, omdat de community erg belangrijk is voor wanneer er informatie of hulp gezocht moet worden, en in de afgelopen jaren heeft Python Java ingehaald in community activiteit.

9.3.3 Structuur



Figuur 9 Class diagram desktop applicatie

Classes

In Figuur 9 is een class diagram te zien van alle classes die gebruikt worden binnen de applicatie.

SoundProcessingCoordinator

SoundProcessingCoordinator is de class waarin alles aan elkaar gekoppeld wordt, en de applicatie gedraaid kan worden. Hierin kan de automatische kalibratie gedraaid worden, kan input van de gebruiker verwerkt worden, is de seriële verbinding geregeld en wordt de ruwe data ontcijferd.

Mic en MicPair

De Mic class is een class bedoeld om de coördinaten van een microfoon te beheren. De MicPair class wordt gebruikt om de afstand tussen twee microfoons en het midden tussen die afstand op te slaan.

Sound

Wanneer een geluid is gedetecteerd, wordt een Sound object aangemaakt. Deze class beheert, normaliseert en analyseert de data van het gemaakte geluid. Dit bevat onder andere de data van het geluid bij alle microfoons.

LocationCalculator

De LocationCalculator is een abstracte class. Deze class heeft maar een (abstracte) methode genaamd run_localization. Beide SrpPhatLocationCalculator en TriangulationLocationCalculator erven van LocationCalculator, en implementeren deze methode op hun eigen wijze, met hun eigen lokalisatiealgoritme. Hierdoor is het makkelijk gemaakt om in de SoundProcessingCoordinator de lokalisatiemethode, zonder de applicatie opnieuw op te hoeven starten, uit te wisselen.

ResultProcessingService

De ResultProcessingService is een class die de verantwoordelijkheid heeft van grafiekjes maken en data wegschrijven naar bestanden. Hiervoor worden drie verschillende methoden gebruikt. De eerste voor het opslaan van de setup data, zoals Mics, MicPairs en geluidssnelheid. De tweede voor het plotten en opslaan van de data uit de Sound class. De laatste voor het plotten en opslaan van de uitslag van de lokalisatie.

Design flaw

Tijdens de implementatie van het POC is er niet goed genoeg nagedacht over een aantal software design principes. Een goed voorbeeld hiervan is het “Single Responsibility Principle”. Dit principe schrijft voor dat elke class die gebruikt wordt in een project niet meer dan een verantwoordelijkheid heeft (Despoudis, 2017). Een goed voorbeeld hiervan is de abstracte class LocationCalculator. Het enige wat die class, en de classes die er van erven, als doel hebben is het bepalen van een locatie. De class heeft geen andere verantwoordelijkheden, en voldoet dus aan het “Single Responsibility Principle”.

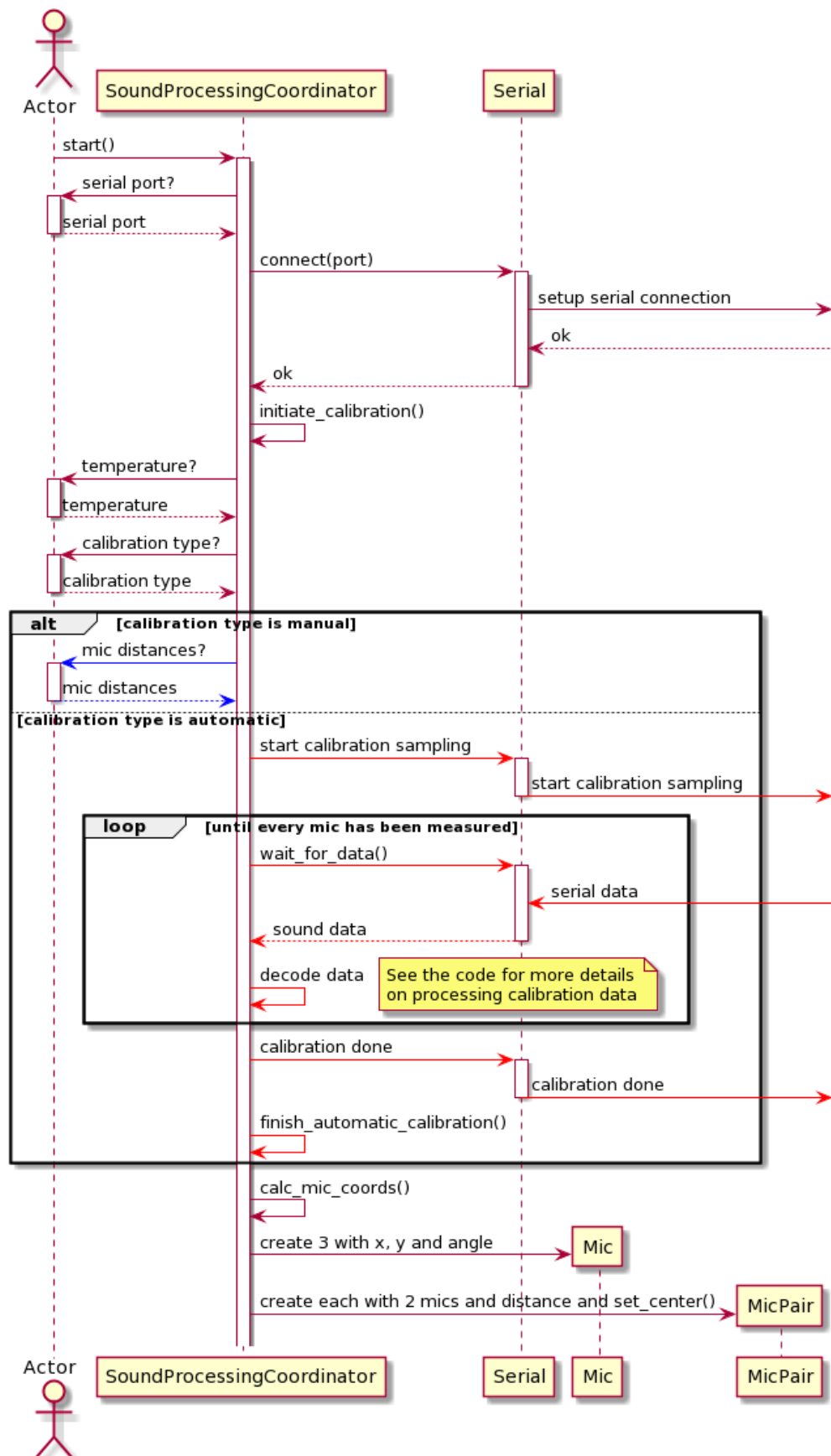
Een voorbeeld van een class waarbij dit principe niet is toegepast, is de SoundProcessingCoordinator. Deze class heeft een hele hoop met verantwoordelijkheden, waaronder kalibratie, de seriële communicatie, het verwerken van ruwe data en het vragen van input aan de gebruiker. Dit zijn te veel verantwoordelijkheden die bij het bedenken en het implementeren van het POC al toegewezen zijn aan deze class. De verantwoordelijkheden zijn wel opgedeeld in functies, maar de class is uiteindelijk niet verder opgesplitst. De verantwoordelijkheden van de SoundProcessingCoordinator class kunnen bijvoorbeeld opgesplitst worden in:

- Een 'State' (benaming kan anders) Singleton class, die een state bijhoudt van wat de instellingen zijn zoals Mic locaties, temperatuur en welke LocationCalculator er gebruikt wordt.
- Een abstracte 'Calibrator' class, die het kalibratieproces regelt, en 'AutomaticCalibrator' en 'ManualCalibrator' die ervan erven;
- Een Abstracte 'Decoder' class die het ontcijferen van de ruwe data op zich neemt. Hierbij kan een specifieke 'SerialSoundDecoder' van de 'Decoder' erven, met de mogelijkheid om uit te kunnen breiden door meer vergelijkbare classes aan te kunnen maken;
- Een 'ConnectionManager' class die alle communicatie met connecties regelt, zoals de seriële verbinding;
- Een abstracte 'Sampler' class, die doorlopend wacht op data van een bepaalde externe verbinding, die verkregen wordt via de ConnectionManager, en die vervolgens verder laat afhandelen door een 'Decoder'. Hier bij kan bijvoorbeeld een specifieke 'SerialSampler' class gebruikt worden;
- Een 'UserInput' class die alle input van de gebruiker verwerkt;
- Een 'Main' class die er voor zorgt dat de applicatie gestart kan worden, en alles samenwerkt.

9.3.4 Sequence diagrams

Setup en kalibratie

In Figuur 10 is een sequence diagram te zien die beschrijft hoe de setup verloopt, en wat er gebeurt bij de kalibratie van de desktop applicatie. De Actor in dit diagram staat voor de gebruiker van de applicatie. Elke keer dat er verwezen wordt naar de Actor, wordt er verwezen naar de output en input van de terminal. Serial is een class geïmporteerd uit de package PySerial. Deze class regelt communicatie van en naar seriële poorten. Wanneer er een seriële poort geselecteerd is, zal er geprobeerd worden mee te verbinden. Dit wordt herhaald totdat het lukt om met de geselecteerde poort te verbinden. Bij de automatische kalibratie wordt er elke keer wanneer het fout gaat, opnieuw een automatische kalibratie geïnitieerd, totdat het goed gaat. Aan het eind van de kalibratie, automatisch of handmatig, worden de coördinaten van de microfoons berekend aan de hand van de afstanden tussen de microfoons. Hiervan worden vervolgens Mic en MicPair objecten gemaakt, die verder in de applicatie gebruikt worden.



Figuur 10 Sequence diagram setup en kalibratie

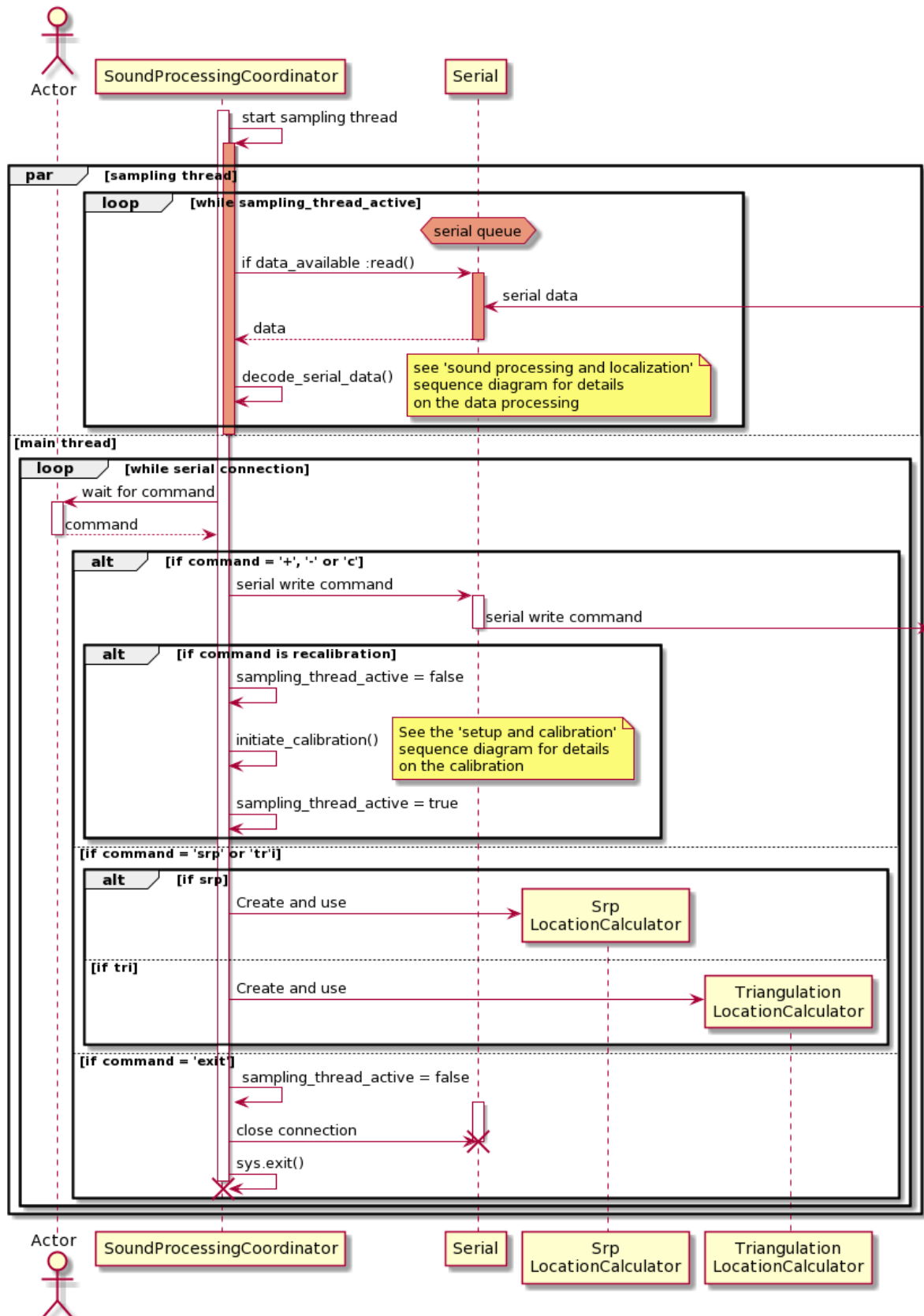
Sampling thread en input

In Figuur 11 is een sequence diagram te zien die laat zien hoe er om gegaan wordt met input van de gebruiker, terwijl er parallel gesampled wordt voor seriële data. Deze sequence diagram beschrijft wat er gebeurt wanneer de kalibratie klaar is.

De input van de gebruiker wordt verwerkt in de main-thread terwijl er in een aparte sampling-thread omgegaan wordt met de data die via de seriële poort ontvangen wordt. De commando's die de gebruiker kan invoeren zijn:

- '+' om de detectie threshold op te hogen, zodat er minder snel een geluid gedetecteerd wordt;
- '-' om de detectie threshold op te hogen, zodat er eerder een geluid gedetecteerd wordt;
- 'c' om opnieuw de kalibratie stappen door te lopen;
- 'srp' om SRP-PHAT in te stellen als lokaliseringmethode;
- 'tri' om triangulatie in te stellen als lokaliseringmethode;
- 'exit' om de applicatie af te sluiten.

Wat er precies gebeurt in de sampling-thread wordt beschreven bij [Processing en lokalisatie](#).



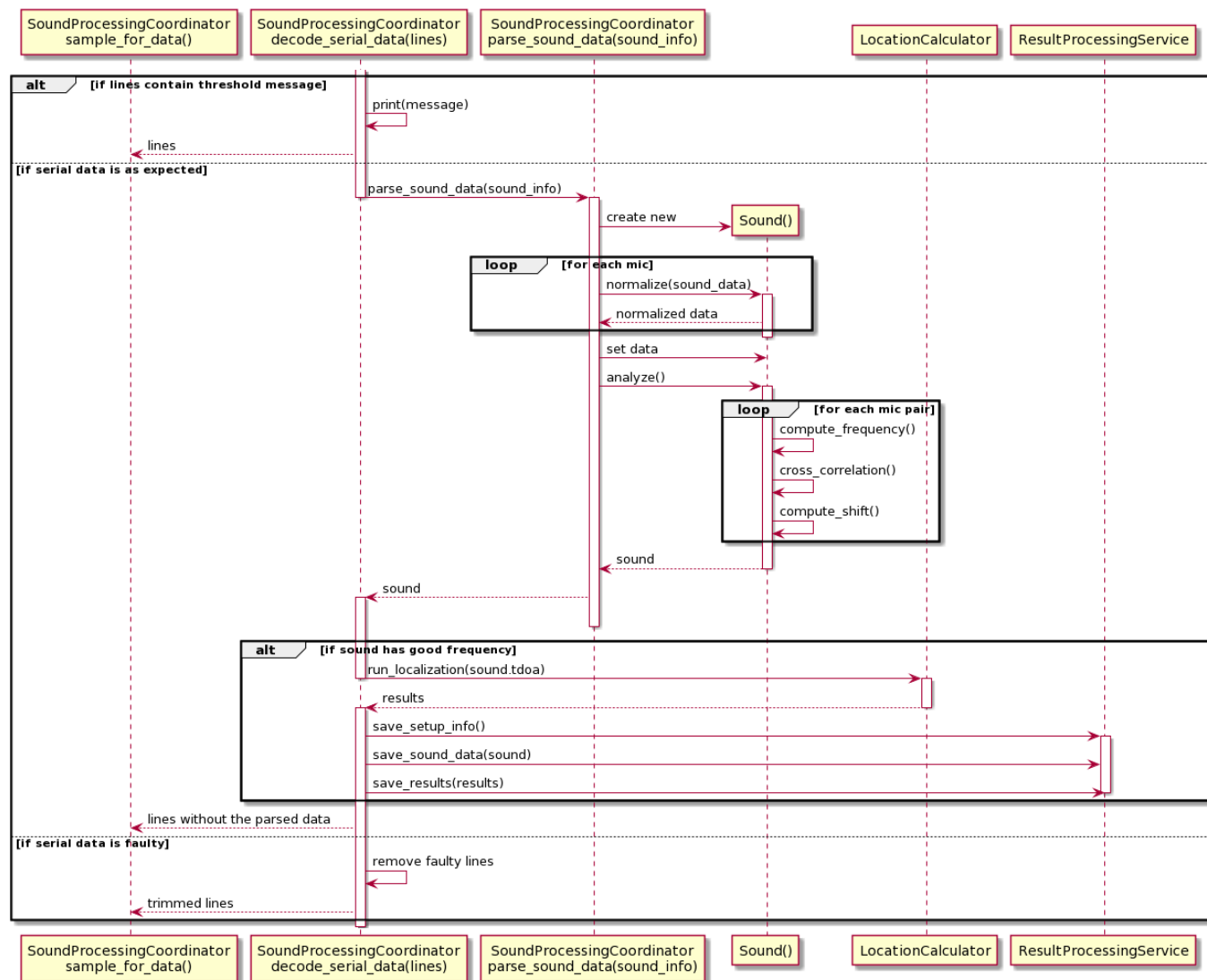
Figuur 11 Sequence diagram threading en input

Processing en lokalisatie

In Figuur 12 is een sequence diagram te zien die het proces laat zien van de sampling-thread. Om te beginnen wordt er gekeken of de regels die gelezen worden uit de seriële verbinding wel data bevatten in het format zoals verwacht wordt. Als dit niet zo is worden er een regels verwijderd tot de regels weer conform het protocol zijn. Als de eerste regel een threshold bericht is, wordt deze naar de terminal gecommuniceerd.

Wanneer er geluidsdata ontvangen is, wordt er een nieuw Sound object aangemaakt. Vervolgens wordt de geluidsdata van elke microfoon genormaliseerd. Wanneer de geluidsdata geanalyseerd wordt, worden de frequenties, cross-correlaties en TDOA's berekend van elk microfoon paar.

De frequenties die bepaald zijn worden gebruikt om te bepalen of het geluid relevant genoeg is om een locatiebepaling te doen. Dit gebeurt alleen wanneer er bij elke microfoon een frequentie berekend is tussen de 500 en 4000Hz. Wanneer dit het geval is, wordt de lokalisatie uitgevoerd en worden de resultaten van de setup, het geluid en de lokalisatie opgeslagen.



Figuur 12 sequence diagram processing en lokalisatie

9.3.5 Performance

De performance is van belang bij het berekenen van een locatie in realtime. Hierdoor moet er bij het schrijven van code extra opgelet worden op langdurende en dure operaties. Ook zal er rekening gehouden worden met memory gebruik wanneer grote hoeveelheden data verwerkt worden.

Algoritme performance

Omdat er SRP-PHAT gebruikt wordt als lokalisering methode, is het van belang dat het systeem nog steeds realtime kan blijven werken.

Omdat er bij elke berekening van SRP-PHAT veel punten moeten berekend worden, vooral als er een voorspelling gedaan moet worden in een grotere range, is er voor gekozen om een variant te implementeren waardmee de rekenkracht die nodig is verkleind wordt.

Van tevoren is een “range” (in meters) en een “initial accuracy” nodig. Er wordt dan op elk punt binnen de aangegeven range, in stappen van de “initial accuracy”, berekend wat de RMSE is van elk punt.

Omdat dit een resource intensief proces is, is het algoritme een klein beetje aangepast. Wanneer alle punten bepaald zijn, zal er om de 10 best geschikte locaties heen, nogmaals het algoritme uitgevoerd worden, maar dan met 1 decimaal meer. Dit proces wordt zo vaak herhaald als dat er aangegeven wordt door de variabele “precision index”, die bij de aanroep van de functie als optionele parameter mee gegeven kan worden.

Om het algoritme ook efficiënter te laten zijn is er gekozen om het yield keyword te gebruiken om een dataframe samen te stellen. Uit de officiële python documentatie

Yield expressions and statements are only used when defining a generator function, and are only used in the body of the generator function. Using yield in a function definition is sufficient to cause that definition to create a generator function instead of a normal function. (Python Software Foundation, 2020)

En

A function which returns a generator iterator. It looks like a normal function except that it contains yield expressions for producing a series of values usable in a for-loop or that can be retrieved one at a time with the next() function. (Python Software Foundation, 2020)

Omdat een dataframe een Iterable aanvaard als parameter, is het dus valide om een generator functie mee te geven, en zo een dataframe te creëren. Hierdoor wordt significant minder RAM geheugen gebruikt dan bij het gebruiken van een list omdat de resultaten direct toegevoegd worden aan het dataframe, en is het algoritme sneller geworden.

Threading

Om ervoor te zorgen dat er input van de gebruiker verwerkt kan worden en er tegelijk data verwerkt kan worden, wordt er gebruik gemaakt van threading. Het proces dat alle

dataverwerking doet wordt in een thread geïnitieerd zodat de input van de gebruiker in de main-thread verwerkt kan worden. Aan de hand van een threading event kan de data sampling thread op de hoogte gesteld worden om te stoppen. Met behulp van dit event en error handling is de applicatie thread safe gemaakt.

9.3.6 Code conventies

Het schrijven van de Python code, de conventies die gebruikt zijn en de standaarden die zijn aangehouden zijn grotendeels terug te vinden in “The Hitchhikers Guide to Python” (Kenneth Reitz, 2011-2020) en de PEP-8 Python style guide (Reitz, 2020).

Linting

Om er voor te zorgen dat de code kwaliteit hoog gehouden wordt, is er gebruik gemaakt van een python linting extension in Visual Studio Code. Zo wordt er tijdens het schrijven van code al gecheckt op kwaliteit en stijl.

9.3.7 Unittesten

Om er voor te zorgen dat de code goed blijft werken, zijn er unittesten geschreven. Deze testen zijn in de broncode terug te vinden met het voorvoegsel “test”. Wanneer er een library gebruikt wordt voor een bepaalde taak, wordt de functionaliteit van die library niet getest. Dit wordt expres niet gedaan, omdat de libraries die gebruikt worden allemaal bekende libraries zijn die hun eigen testen hebben, of een deel is van de Python standard library. Soms wordt wel getest of deze libraries correct geïmplementeerd worden. Meer over de unittesten is te vinden in de bijlagen bij [16.2 unittesten](#).

10 Systeemtesten

Elke test zal gedaan worden in een bepaalde setting. Voor elke testsetting combinatie die er is zullen er op verschillende vooraf gedefinieerde punten geluid gemaakt worden. Er gaat gekeken worden tot op welke afstanden het systeem accuraat blijft. Er zal vergeleken worden of automatische kalibratie beter of minder goed werkt dan het zelf opmeten en invullen van de data.

10.1 Testen uitvoeren

Voor elke test zullen er een aantal karakteristieken genoteerd worden:

- Tijd van berekening
- Omgeving
 - o Weersomstandigheden (windsnelheid en temperatuur)
 - o Obstakels
 - o Muren/grond/plafond
- Afstand tussen de microfoons
- De plaats van het geluid
- De voorspelling van de plaats van het geluid
 - o Exact berekend punt
 - o Het verschil met de echte plaats

Het is belangrijk om bij het testen zo veel mogelijk informatie te noteren over de omstandigheden. Wanneer er zich iets voor doet wat niet verwacht was, zoals een reeks metingen die afwijken van wat verwacht wordt, kan er gekeken worden waardoor dat mogelijk zou kunnen komen. Het grootste gedeelte van de ruwe data is te vinden in de bijlagen, in het mapje 'results'. Hoe deze data te interpreteren is, kan je vinden in paragraaf [17.1 bijlagen: resultaten overige](#).

10.2 Scenario's

Om te zorgen dat duidelijk is waar met welke variabelen getest wordt, zijn hier de details van de testscenario's omschreven.

10.2.1 Scenario 1

In Figuur 13 is een plattegrond te zien met een aantal obstakels die tijdens de metingen aanwezig waren. **Error! Reference source not found.** en Figuur 23 in de bijlagen is zijn foto's te zien van de omgeving tijdens de metingen. In Figuur 24 is een plattegrond te zien van het gebouw waarbij getest is. Er zal in dit scenario uit gegaan worden van dat er weinig of geen geluid van binnen het gebouw zal komen. Het rekening houden met extreme weersomstandigheden en neerslag zit buiten scope van dit project. Er is voor deze locatie gekozen omdat het relatief dichtbij ALTEN ligt, en de mogelijkheid er is om op deze locatie tijdens de onderzoeksperiode metingen te verrichten. Ook is hier voor gekozen omdat er een aantal obstakels en omgevingsvariabelen een rol spelen, wat interessant is om te onderzoeken.

De afstanden tussen de microfoons waren zijn volgt:

- mic 0 tot mic 1: 4.2 meter;
- mic 0 tot mic 2: 6.45 meter;
- mic 1 tot mic 2: 4.9 meter.

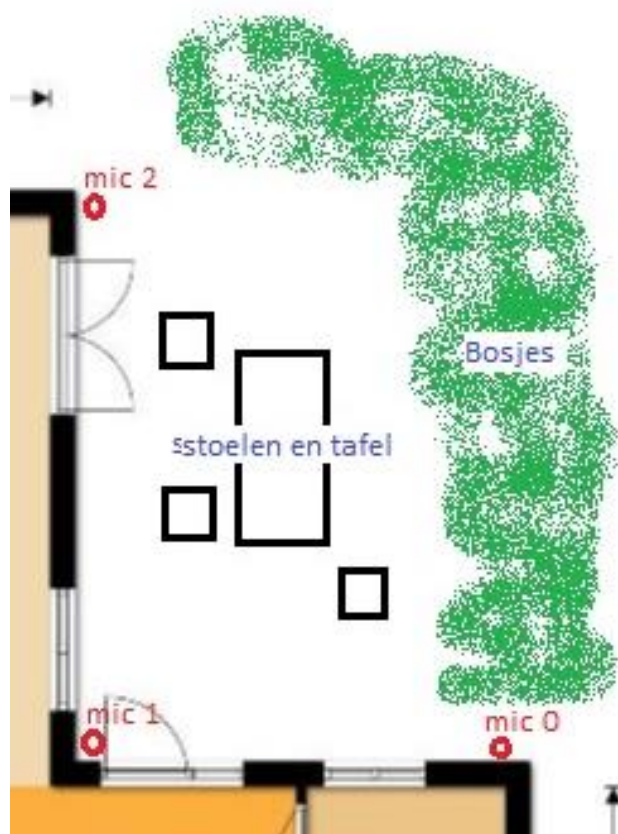
De microfoons zaten ongeveer 20cm van de grond aan regenpijpen, en allemaal ongeveer op dezelfde hoogte. Dit is zo gedaan omdat er op die hoogte bij elke regenpijp een verbinding met de muur erachter was, en het makkelijk was om de microfoons zo op een gelijke hoogte te krijgen. Het is belangrijk om bij een 2D lokalisering de microfoons op een gelijke hoogte te hebben, omdat er geen rekening gehouden wordt met hoogteverschillen tussen de microfoons in een 2D lokalisering.

Temperatuur: tussen 17 en 19°C

Windsnelheid: tussen 3.9 en 5 m/s

Obstakels: de grond, de muren achter de microfoons, een aantal stoelen, tafels en struikjes

Datum: 28/05



Figuur 13 Scenario 1 plattegrond

10.2.2 Scenario's 2 en 3

Scenario 2 en 3 zijn allebei binnen op het kantoor van ALTEN in Apeldoorn gemeten, met vergelijkbare omgevingsvariabelen. In Figuur 14 is een plattegrond te zien die een overzicht geeft van de ruimte waarin getest wordt. In Figuur 18 en Figuur 21 in de bijlagen is te zien van hoe de omgeving er in het echt uit zag.

In scenario 2 waren de afstanden tussen de microfoons:

- mic 0 tot mic 1: 3.86 meter;
- mic 0 tot mic 2: 5.94 meter;
- mic 1 tot mic 2: 4.66 meter.

In scenario 3 zijn de afstanden tussen de microfoons:

- mic 0 tot mic 1: 3.86 meter;
- mic 0 tot mic 2: 2.85 meter;
- mic 1 tot mic 2: 2.70 meter.

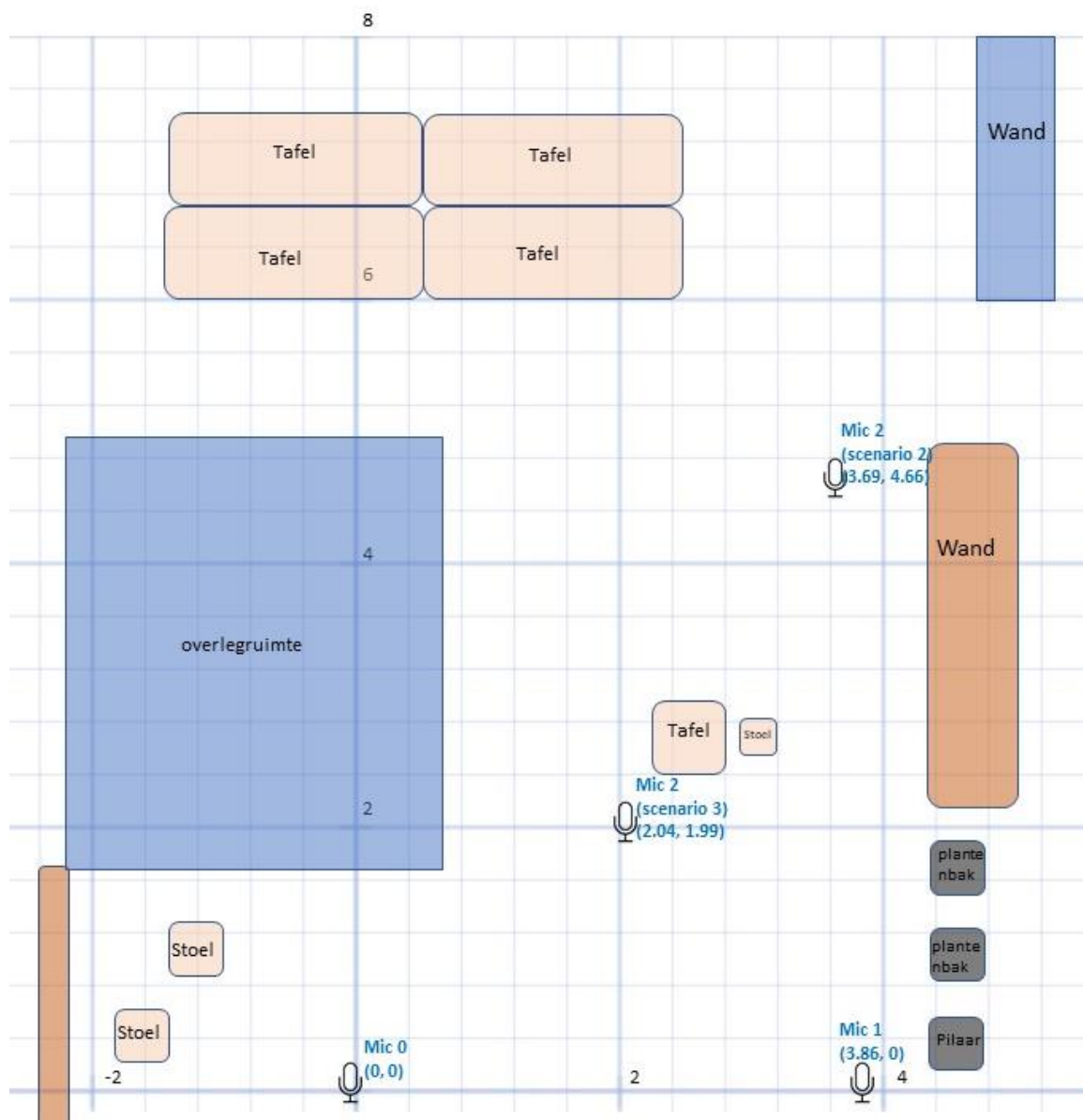
Temperatuur: tussen 22 en 24°C

Windsnelheid: Geen, want het is binnen

Obstakels: de muren en wanden in de ruimte, een tafel en stoel, een glazen wand aan de zijkant, de grond en het plafond

Datum: 30/05

Etc.: de microfoons zaten ongeveer 15 cm van de grond aan koffiekannen gebonden die gevuld waren met water. Dit is zo gedaan omdat de koffiekannen een stevige basis boden en de microfoons op dezelfde hoogte opgezet konden worden. Het is belangrijk om bij een 2D lokalisering de microfoons op een gelijke hoogte te hebben, omdat er geen rekening gehouden wordt met hoogteverschillen tussen de microfoons.



Figuur 14 abstracte plattegrond scenario's 2 en 3

11 Resultaten

Om het proof of concept te kunnen valideren, moeten er twee aspecten gepeild worden, de automatische kalibratie en de locatie bepaling.

11.1 Automatische kalibratie

Bij de automatische kalibratie is er getest in drie scenario's, zoals beschreven in [hoofdstuk 10.3 Scenario's](#). De automatische kalibratie testen zijn uitgevoerd door een aantal stappen door te lopen, zoals beschreven bij [paragraaf 7.2.5 deelvraag 5 kalibratie](#) bij het onderdeel automatische kalibratie.

11.1.1 Scenario 1

Ruwe metingen

AK staat voor automatische kalibratie. In de vakjes is de gemeten afstand tussen de twee microfoons van het microfoonpaar te zien. Van links naar rechts worden de 3 verschillende microfoonparen bekeken. Dit staat dus voor de afstand tussen de twee microfoons.

De echte waarden van de afstand tussen de microfoons is er bij gezet om een duidelijker beeld te geven waarmee de afstanden vergeleken kunnen worden.

Test ▼ /mic paar(m) ►	Mic 0 – mic 1	Mic 0 – mic 2	Mic 1- mic 2
Echte waarden	4.2	6.45	4.9
AK 1	3.146	6.088	4.139
AK 2	2.274	6.152	3.890
AK 3	3.105	5.681	4.385
AK 4	3.711	6.518	3.958
AK 5	3.446	6.786	4.020

Tabel 6 Ruwe metingen automatische kalibratie scenario 1

De ruwe waarden schetsen maar een gelimiteerd beeld van wat de data echt in houdt. Daarom is er zijn de waarden van de testen in Tabel 7 omgezet naar het verschil tussen de lengte van de berekende en echte waarden.

Test ▼ /mic paar verschil (m/%) ►	Mic 0 – mic 1	Mic 0 – mic 2	Mic 1- mic 2	RMSE
AK 1	1.054/25.1%	0.362/5.6%	0.761/15.5%	0.779/17.3%
AK 2	1.926/45.9%	0.298/4.6%	1.010/20.6%	1.267/29.2%
AK 3	1.095/26.1%	0.769/11.9%	0.515/10.5%	0.828/17.6%
AK 4	0.489/11.6%	0.068/1.1%	0.942/19.2%	0.614/13.0%
AK 5	0.754/18.0%	0.336/5.2%	0.880/18.0%	0.697/15.0%
RMSE	1.168/27.8%	0.431/6.7%	0.840/17.1%	

Tabel 7 Verschil in afstand tegenover echte waarden met een RMSE

Het gemiddelde verschil tussen de echte afstanden, en de berekende afstanden is 0.75 meter. Het gemiddelde procentuele afstandsverschil is 15.9%.

11.1.2 Scenario 2

In scenario 2 is er geconstateerd dat er, met dezelfde afstanden als wanneer er buiten gekalibreerd werd, het niet mogelijk was om de automatische kalibratie te voltooien. Wanneer er veel weerkaatsing van de omgeving voorkomt, zal het cross correlation algoritme dat ook meenemen in de berekening van de TDOA's. Hierdoor kan het zich voorkomen dat de berekende TDOA's erg afwijken van de werkelijke TDOA's. De manier om de automatische kalibratie te doen meet de afstanden tussen de microfoons altijd vanuit twee TDOA's, en vergelijkt vervolgens of deze tot op zekere hoogte overeen komen. Bij scenario 2 bleek dit nooit binnen acceptabele grenzen te zijn, en kon er dus niet automatisch gekalibreerd worden.

11.1.3 Scenario 3

Omdat er bij scenario 2 geen automatische kalibratie mogelijk was, is er gekozen voor een alternatieve setup, in dezelfde ruimte, waar hopelijk minder weerkaatsing van het geluid bij zou voorkomen. Door de gelimiteerde ruimte die beschikbaar is bij deze setup gekozen voor een kleinere setup. Helaas bleek hetzelfde probleem ook op te spelen in dit scenario. Het resulteerde in vergelijkbare resultaten van de kalibratie als bij scenario 2, waarbij er geen overeenkomende TDOA's voorkwamen tijdens de automatische kalibratie, waardoor deze fout bleef gaan.

11.2 Locatiebepaling

Voor de locatiebepaling zijn er 3 verschillende test scenario's. De metingen zijn uitgevoerd door meerdere keren op de van tevoren bedachte locatie in relatie tot de microfoons te klappen. Wanneer er vervolgens een geluid gedetecteerd werd die geaccepteerd werd (goede frequentie waarden van de verschillende geluiden) werd dit resultaat opgeslagen en werd er doorgedaan naar de volgende locatie. De locatie waar geklapt is, is een schatting van de coördinaten, gemeten met een meetlint.

11.2.1 Scenario 1

Dit zijn de testresultaten van de testen gedaan in scenario 1. Scenario 1 staat beschreven in [Hoofdstuk 10.3.1](#).

Bij het testen van de locatie bepaling is er, zoals hierboven beschreven, geklapt bij een aantal coördinaten in de omgeving van de microfoons. In Tabel 8 staan de testresultaten. Van links naar rechts zijn oplopende x coördinaten te zien van alle plekken waar geklapt is. Van boven naar beneden zijn de oplopende y coördinaten te zien van alle plekken waar geklapt is. Elk vakje representeert een set x, y coördinaten. De waarden die in de vakjes staan zijn de coördinaten die door de applicatie voorspeld zijn.

Y(m)▼/X(m)►	-3	-1	1	2	3	4
0		X: -1.69 Y: 0.08	X: 1.04 Y: 0.00		X: 2.87 Y: 0.13	
1		X: -1.10 Y: 0.00	X: 1.02 Y: 1.41		X: 5.35 Y: 1.66	
2						
3	X: -2.04 Y: 2.64	X: -0.53 Y: 2.71	X: 1.29 Y: 2.60		X: 2.73 Y: 2.25	
5	X: -5.33 Y: 8.19	X: -1.22 Y: 6.02	X: 1.25 Y: 5.24		X: 3.55 Y: 4.89	
7		X: -0.09 Y: 2.40	X: 2.23 Y: 5.71		X: 3.50 Y: 6.35	

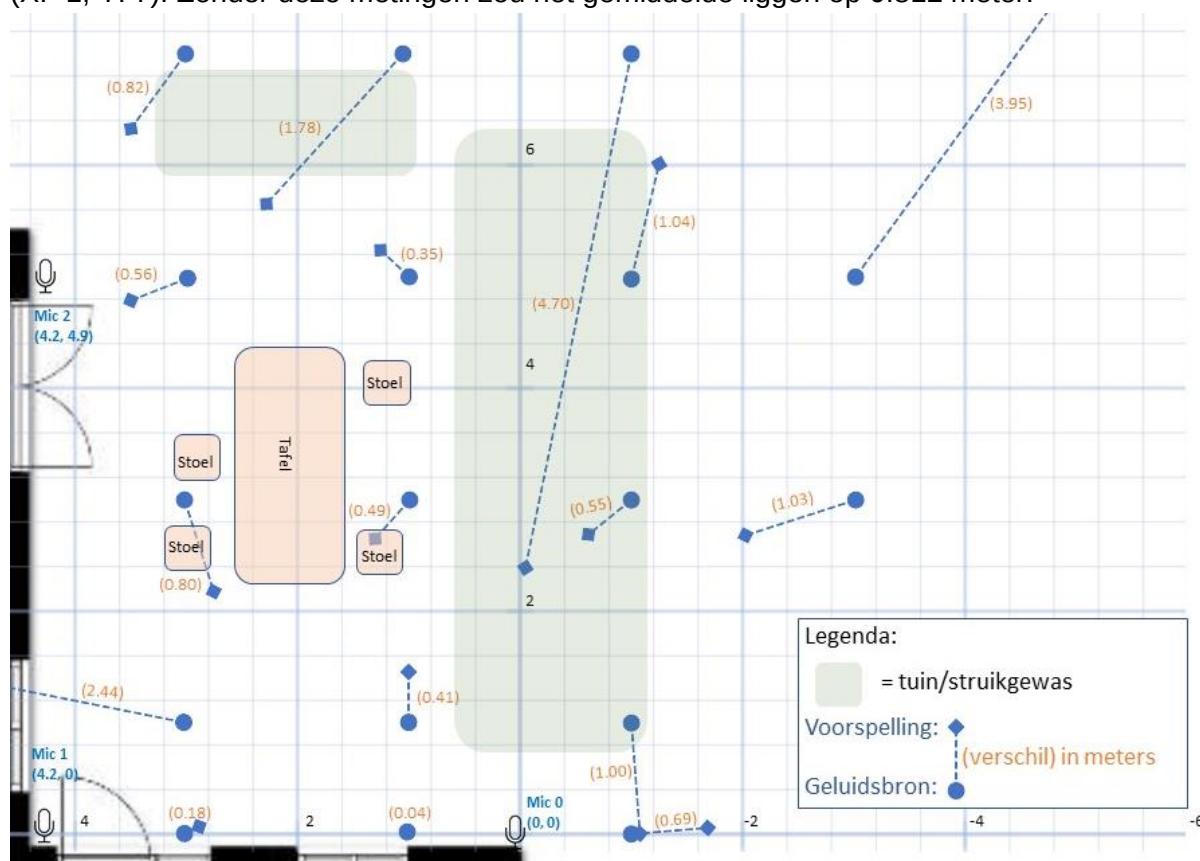
Tabel 8 Coördinaten echt vs berekend scenario 1

In Tabel 9 is de data gerepresenteerd als het verschil in afstand tussen het gemeten punt en de echte waarde.

Y(m)▼/X(m)►	-3	-1	1	2	3	4
0		0.69	0.04		0.18	
1		1.00	0.41		2.44	
2						
3	1.03	0.55	0.49		0.80	
5	3.95	1.04	0.35		0.56	
7		4.70	1.78		0.82	

Tabel 9 Verschil in afstand metingen met echte waarden scenario 1

De gemiddelde fout van de metingen is 1.23 meter. Dat dit gemiddelde zo hoog is, komt vooral door een aantal metingen die op een grotere afstand gedaan zijn, op (X: -3, Y: 5) en (X: -1, Y: 7). Zonder deze metingen zou het gemiddelde liggen op 0.812 meter.



Figuur 15 Metingen scenario 1 lokalisatie met SRP-PHAT op plattegrond van omgeving

Elk blauw rond puntje dat te zien is op Figuur 15, is een plaats waar geklapt is. Elk blauw ruitje representeert een lokalisatie bepaald door het POC. De lijn tussen de twee is het afstandsverschil tussen de voorspelde en de echte locatie van de geluidsbron. Op de plattegrond is ook te zien waar de microfoons zich bevinden bij dit scenario. Wat er tussen de haakjes staat bij de microfoons zijn de x en y coördinaten van elke microfoon. Omdat de omgeving waarin gemeten is belangrijk is voor de resultaten, en invloed kan hebben op de metingen, zijn de plekken van obstakels en dergelijke aangeduid met de doorzichtige vlakken.

11.2.2 Scenario 2

Dit zijn de testresultaten van de testen gedaan in scenario 2, beschreven in [Hoofdstuk 10.3.2](#).

Ruwe metingen

Bij het testen van de locatie bepaling is er, zoals hierboven beschreven, geklapt bij een aantal coördinaten in de omgeving van de microfoons. Van links naar rechts zijn oplopende x coördinaten te zien van alle plekken waar geklapt is. Van boven naar beneden zijn de oplopende y coördinaten te zien van alle plekken waar geklapt is. Elk vakje representeert een set x, y coördinaten. De waarden die in de vakjes staan zijn de coördinaten die door de applicatie voorspeld zijn.

Y(m)▼/X(m)►	-3	-1	1	2	3	4
0		X: 0.07 Y: 0.43	X: 1.05 Y: -0.34		X: 2.85 Y: -0.06	
1			X: 1.68 Y: 0.58		X: 3.01 Y: 1.37	
2						
3			X: 1.15 Y: 2.59		X: 2.80 Y: 2.08	
5			X: 1.99 Y: 2.24		X: 2.57 Y: 2.13	
7						

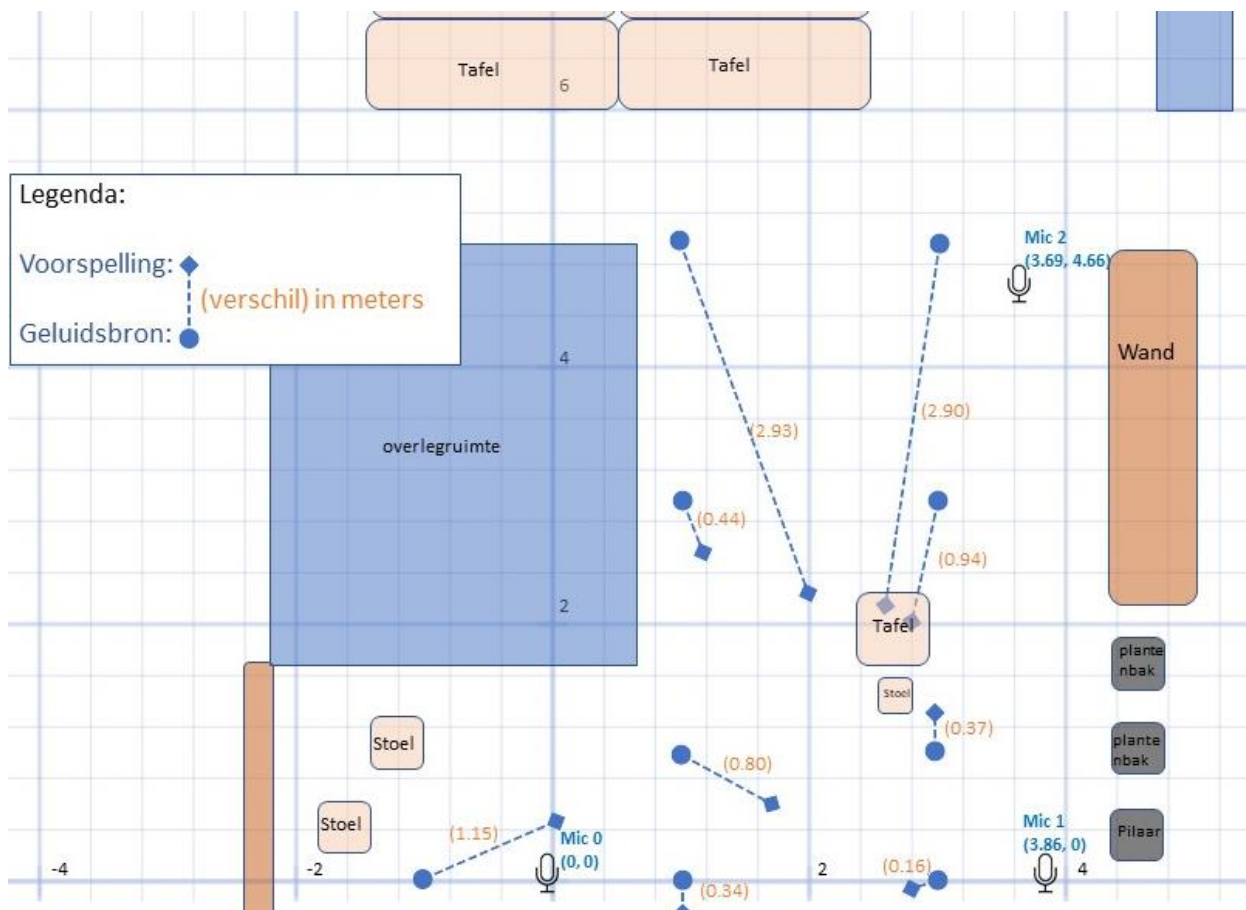
Tabel 10 Coördinaten echt vs berekend scenario 2

In Tabel 11 is de data gerepresenteerd als het verschil in afstand tussen het gemeten punt en de echte waarde.

Y(m)▼/X(m)►	-3	-1	1	2	3	4
0		1.15	0.34		0.16	
1			0.80		0.37	
2						
3			0.44		0.94	
5			2.93		2.90	
7						

Tabel 11 Verschil in afstand metingen met echte waarden scenario 2

De gemiddelde fout van de metingen is 1.11 meter. Dit komt vooral door de metingen die op grotere afstanden gedaan zijn. Zonder de twee metingen die gedaan zijn bij Y=5, is de gemiddelde fout 0.6 meter.



Figuur 16 Metingen scenario 2 lokalisatie met SRP-PHAT op plattegrond van omgeving

Elk blauw rond puntje dat te zien is op Figuur 16 is een plaats waar geklapt is. Elk blauw ruitje representeert een lokalisatie bepaald door het POC. De lijn tussen de twee is het afstandsverschil tussen de voorspelde en de echte locatie van de geluidsbron. Op de plattegrond is ook te zien waar de microfoons zich bevinden bij dit scenario. Wat er tussen de haakjes staat bij de microfoons zijn de x en y coördinaten van elke microfoon. Omdat de omgeving waarin gemeten is belangrijk is voor de resultaten, en invloed kan hebben op de metingen, zijn de plekken van obstakels en dergelijke aangeduid met de doorzichtige vlakken.

11.2.3 Scenario 3

Dit zijn de testresultaten van de testen gedaan in scenario 2, beschreven in [Hoofdstuk 10.3.2](#).

Ruwe metingen

Bij het testen van de locatie bepaling is er, zoals hierboven beschreven, geklapt bij een aantal coördinaten in de omgeving van de microfoons. Van links naar rechts zijn oplopende x coördinaten te zien van alle plekken waar geklapt is. Van boven naar beneden zijn de oplopende y coördinaten te zien van alle plekken waar geklapt is. Elk vakje representeert een set x, y coördinaten. De waarden die in de vakjes staan zijn de coördinaten die door de applicatie voorspeld zijn.

Y(m)▼/X(m)►	-3	-1	1	2	3	4
0		X: 0.59 Y: 0.53		X: 1.24 Y: 0.44	X: 2.67 Y: 0.17	
1				X: 2.23 Y: 0.98		
2						X: 3.81 Y: 1.82
3						
5						
7						

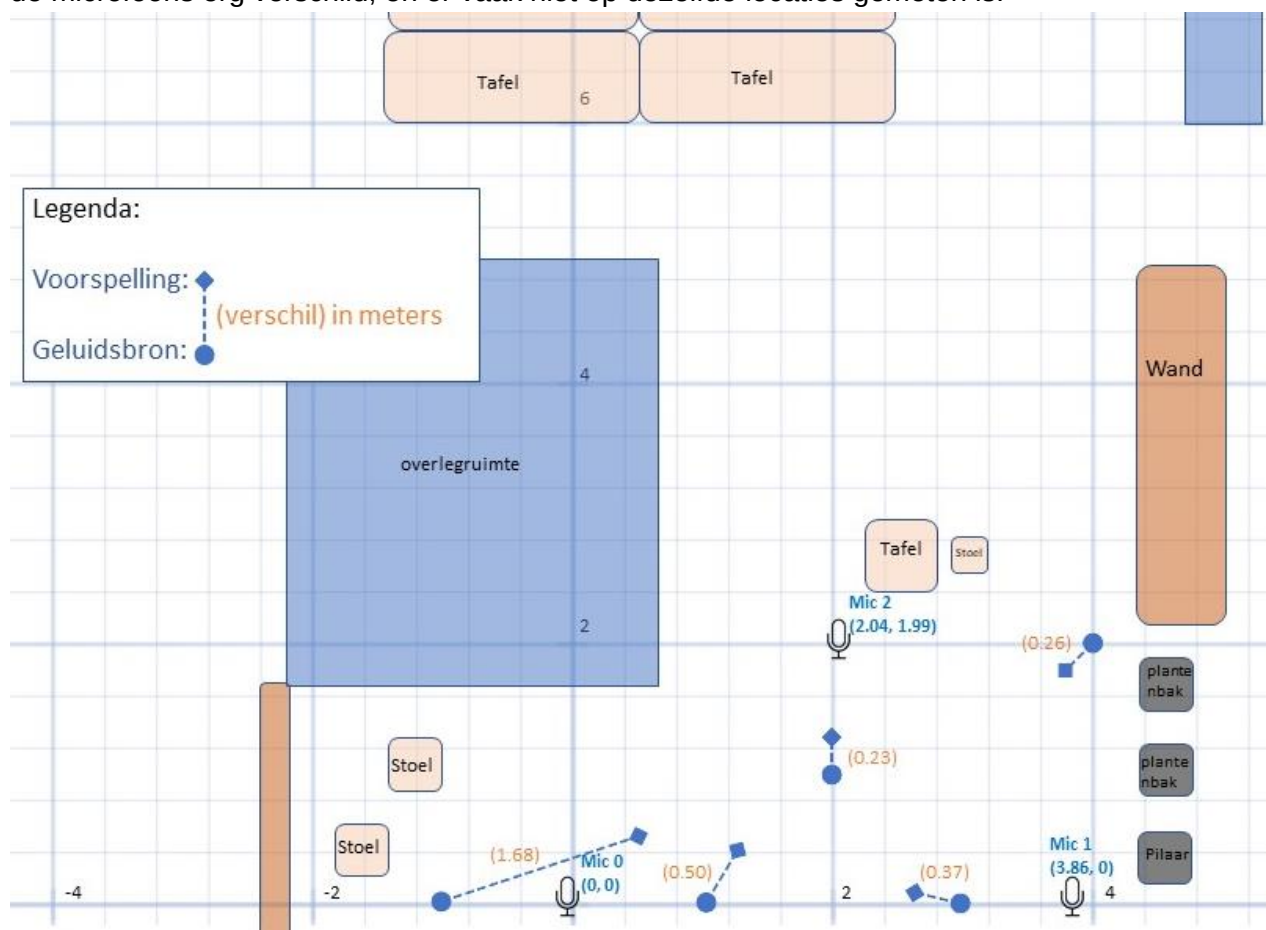
Tabel 12 Coördinaten echt vs berekend scenario 3

In is de data gerepresenteerd als het verschil in afstand tussen het gemeten punt en de echte waarde.

Y(m)▼/X(m)►	-3	-1	1	2	3	4
0		1.68	0.50		0.37	
1				0.23		
2						0.26
3						
5						
7						

Tabel 13 Verschil in afstand metingen met echte waarden scenario 3

De gemiddelde fout van de metingen is 0.608 meter. Het is lastig om deze metingen te vergelijken met de metingen van de andere twee scenario's, omdat de afstanden van tussen de microfoons erg verschillend, en er vaak niet op dezelfde locaties gemeten is.



Figuur 17 Metingen scenario 3 lokalisatie met SRP-PHAT op plattegrond van omgeving

In Figuur 17, zijn de metingen van de testen die gedaan zijn in scenario 3 op een plattegrond weergegeven.

11.3 Onderzoeksvragen

We kijken nogmaals naar de deelvragen, maar nu met de inzichten die we van de testresultaten gekregen hebben.

Hoeveel invloed heeft de omgeving waarin het systeem geplaatst is op de accuraatheid van de metingen?

Wat vooral op te merken is uit de resultaten van de automatische kalibratie, is dat er in verschillende omgevingen ook totaal andere resultaten zijn gekomen. De invloed van de omgeving was zelfs zo groot, dat er geen automatische kalibratie gedaan kon worden in scenario 2 en 3.

Het valt op dat bij de locatiebepalingen van scenario 1 de metingen die bij Y:5 gedaan zijn, een stuk accurater zijn dan dezelfde metingen bij scenario 2. Door de kleine hoeveelheid meetresultaten is het lastig om te zeggen hoe veel invloed de omgeving precies voor deze metingen heeft gehad, maar er kan geconstateerd worden dat de effectieve meetafstand van scenario 1 groter is dan die van scenario 2.

De resultaten van de kalibratietesten bevestigen de voorspelling dat het meten van consistente resultaten moeizamer is in een gesloten omgeving.

Welke lokaliseringmethode is het meeste geschikt om te gebruiken binnen het project?

Tijdens het onderzoeken van de lokaliseringmethoden, is er vooral gekeken naar de performance in onstuimige omgevingen. Uit dat literatuuronderzoek bleek dat SRP-PHAT hier uitermate geschikt voor is.

Er zijn geen systeemtesten uitgevoerd met het ontwikkelde triangulatie algoritme, omdat het zeer onnauwkeurig is bevonden. Dit is te zien aan de resultaten van de unittesten die geschreven zijn. Deze zijn te vinden in [paragraaf 16.2 unittesten](#). Hierom blijft het antwoord op deze vraag hetzelfde: SRP-PHAT.

Wat voor verschillende variabelen hebben invloed op het bepalen van een locatie van een geluidsbron?

Wat voor invloed de verschillende variabelen precies gehad hebben op de metingen, naast de invloed van de omgeving, is niet helemaal duidelijk geworden door de resultaten van de systeemtesten.

Hoe accuraat kan de locatie van een statische geluidsbron bepaald worden met behulp van geluidsdetectoren?

Wanneer er binnen een beperkt bereik geluid gemaakt wordt kan de locatie van een geluidsbron in scenario 1 bepaald worden met een gemiddelde fout in afstand van 0.812 meter. Wanneer er binnen een beperkt bereik geluid gemaakt wordt kan de locatie van een geluidsbron in scenario 2 bepaald worden met een gemiddelde fout in afstand van 0.6 meter. De locatie van een geluidsbron kan binnen een beperkt bereik in scenario 3 bepaald worden met een gemiddelde fout in afstand van 0.608 meter.

Het gemiddelde berekend over alle metingen is ~1.1 meter.

De meest accurate meting was in scenario 1, met een verschil van 0.04 meter tussen de echte locatie en de berekende locatie. De minst accurate meting was ook in scenario 1, met een verschil van 4.70 meter tussen de echte locatie en de berekende locatie.

Is het mogelijk om met een automatische kalibratie, betere parameters te kunnen verkrijgen voor het bepalen van een locatie van een geluidsbron, dan met een handmatige kalibratie?

Korte antwoord: Theoretisch gezien niet, en praktisch gezien misschien.

Het verschil tussen de echte afstanden, en de berekende afstanden is gemiddeld 0.75 meter. Het gemiddelde procentuele afstandsverschil is 15.9%.

Het feit dat de afstanden tussen de microfoons die berekend worden met de automatische kalibratie afwijken van de waarden die gemeten worden met een meetlint, geeft aan dat het theoretisch gezien beter is om de waarden zelf in te vullen.

De reden dat er praktisch gezien niet een duidelijk antwoord gegeven kan worden, is omdat alleen de waarden van de automatische kalibratie zijn getest, en niet de meetresultaten die zouden komen uit het automatisch gekalibreerde systeem. Om hierop met volle vertrouwen een antwoord op te kunnen geven, moet dit nog getest worden.

12 Conclusie

In dit project is antwoord gezocht op de vraag: **Hoe accuraat is het mogelijk om de realtime locatie van een bewegende geluidsbron te bepalen met behulp van geluidsdetectoren?**

Uit de resultaten van de systeemtesten is gebleken dat het mogelijk is om, met het SRP-PHAT algoritme, tot op 4cm nauwkeurig de locatie van een geluidsbron te kunnen bepalen, wanneer deze meting gedaan wordt in met variabelen en in een omgeving zoals beschreven in [10.3.1 scenario 1](#). Echter de nauwkeurigheid bij de metingen was niet consistent. Gemiddeld was de nauwkeurigheid van het systeem, over alle scenario's, is ~1.1 meter. De afstand waarop gemeten werd bleek van grote invloed te zijn op de precisie van de meting. Op grotere afstanden van de microfoons is er een grotere kans op onnauwkeurige metingen.

Bij de systeemtesten is ook de automatische kalibratie getest. De resultaten van de kalibratietesten bevestigen de hypothese dat het meten van consistente resultaten moeizamer is in een gesloten omgeving, vergeleken met een omgeving in de buitenlucht. De kalibratietesten die in de buitenlucht gedaan zijn resulteerde in een gemiddeld verschil tussen de echte afstanden, en de gemeten afstanden van de kalibratie is 0.75 meter. Het gemiddelde procentuele afstandsverschil is 15.9%.

13 Aanbevelingen

Er zijn een hele hoop scenario's die nog getest kunnen/moeten worden. Onder andere in verschillende scenario's met verschillende microfoon afstanden.

De automatische kalibratie kan mogelijk aangepast worden zodat die ook beter werkt in een afgesloten ruimte. Een van de dingen die hierbij verbeterd kan worden is de bepaling van de TDOA's. Wanneer dit verbeterd is, kunnen de locaties van de geluiden ook beter bepaald worden.

Er zijn naast de verbeteringen van het systeem ook nog heel veel mogelijkheden om uit te breiden op dit onderzoek en het POC. Hier zijn er een aantal:

- Een realtime locatie zou getoond kunnen worden op een plattegrond;
- Meer microfoons in grotere omgeving;
- Koppeling met een home-automation systeem;
- Identificatie en classificatie van geluid.

Er zijn ook bepaalde beveiligingsaspecten die mogelijk interessant zijn. Bijvoorbeeld om inbrekers te detecteren in het donker.

Machine learning of deep learning zijn alternatieven die gebruikt kunnen worden voor de locatiebepaling. Ook kan machine learning gebruikt worden om geluiden te kunnen identificeren. (Wei Ma, 2019) Omdat er voor dit project geen identificatie wordt gedaan, zou dit een goede toevoeging zijn.

Omdat dit project gefocust is op 2D lokalisatie, zijn de 3D lokaliseringsmethoden niet verder behandeld. Als er verder gekeken wordt naar 3D lokalisatie kan de lokaliseringsmethode die nu gebruikt wordt, SRP-PHAT, aangepast worden zodat het ook in 3D werkt. Wat hier voor nodig zou zijn is minimaal 1 microfoon extra, en een aanpassing aan de manier waarop de kalibratie op dit moment gedaan wordt. In plaats van alleen de x-as en y-as te bekijken, moet er dan ook naar de z-as gekeken worden.

14 Nawoord

Als eerste wil ik graag ALTEN bedanken voor het geven van de mogelijkheid om een afstudeerstage te mogen doen bij dit geweldige bedrijf.

Ik wil graag al mijn collega's vanuit ALTEN bedanken.

Merijn Tieltjes als inhoudelijk begeleider.

Hugo Logmans als Technical manager, product-owner en voor het gebruik mogen maken van zijn vakantiehuisje voor het testen van mijn proof of concept.

Sjors van Tongeren als business manager.

Bedankt Jan Oene Krist, Erik Ensing, Rimon Oz en Reinier Kuipers voor de second opinion en de hulp.

Bedankt aan andere afstudeerders die tegelijkertijd begonnen zijn, waar ik waardevolle feedback en handige inzichten van heb gekregen, en waarmee ik met veel plezier gewerkt heb. Daan van den Hoek, Florian Hagens, Gijs Goudzwaard, Wendy Bolt en Casper Uiterwijk.

Als laatste bedankt Eelco Jannink voor de begeleiding vanuit Saxion.

15 Bibliografie

- 3D_sound_localization* - Wikipedia. (2020, march). Opgehaald van Wikipedia:
https://en.wikipedia.org/w/index.php?title=3D_sound_localization&oldid=944405180
- Aviom. (2015). *balanced-vs-unbalanced/*. Opgehaald van Aviom:
<http://www.aviom.com/blog/balanced-vs-unbalanced/>
- BJC. (2017). *Digital Cables and Analog Cables-- whats the difference?* Opgehaald van blue jeans cable: <http://www.bluejeanscable.com/articles/digitalanalog.htm>
- Bracewell, R. (1965). *"Pentagram Notation for Cross Correlation." The Fourier Transform and Its Applications*. New York: McGraw-Hill.
- Chili math. (2011-2020). *distance formula*. Opgehaald van chilimath:
<https://www.chilimath.com/lessons/intermediate-algebra/distance-formula/>
- CUI Devices. (2020, January 17). *DATASHEET MODEL: CMA-4544PF-W | DESCRIPTION: ELECTRET CONDENSER MICROPHONE*. Opgehaald van CUI devices:
<https://www.cuidevices.com/product/resource/cma-4544pf-w.pdf>
- Daddy, F. u. (2016). *Balanced (XLR, TRS) Vs. Unbalanced (RCA, TS) Connection*. Opgehaald van blue-ray.com forum: <https://forum.bluray.com/showthread.php?t=214470>
- Dalskov, D. (2014). *Locating Acoustic Sources with Multilateration*. Aalborg, Denmark: Department of Electronic Systems - Acoustics.
- Despoudis, F. (2017). *Understanding SOLID Principles: Single Responsibility*. Opgehaald van Codeburst: <https://codeburst.io/understanding-solid-principles-single-responsibility-b7c7ec0bf80>
- Eargle, J. (2004). *The Microphone Book*. Elsevier/Focal Press.
- Engineering ToolBox. (2003). *Velocity of Sound in Air*. Opgehaald van Engineering ToolBox:
https://www.engineeringtoolbox.com/air-speed-sound-d_603.html
- Funda hoge bergweg 16. (2020, Februari 13). Opgehaald van Funda:
<https://www.funda.nl/recreatie/verkocht/beekbergen/huis-40801061-hoge-bergweg-16-g201>
- Hannah, L. (2006). *Wind and Temperature Effects on Sound Propagation*. Malcolm Hunt Associates.
- Hoang Do, H. F. (2007). *A real-time srp-phat source location implementation using stochastic region contraction (src) on a large-aperture microphone array*. Opgehaald van CiteSeer:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.219.2770&rank=1>
- Hopper, T. (2010, 02 22). *"How the Clapper Works"*. Opgehaald van HowStuffWorks.com:
<https://home.howstuffworks.com/clapper1.htm>
- hyperphysics. (2007). *Image of speed of sound*. Opgehaald van hyperphysics:
<http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/imgsou/sspe3.gif>
- kaart gemiddelde windsnelheid*. (2020). Opgehaald van klimaatatlas:
<http://www.klimaatatlas.nl/klimaatatlas.php?wel=wind&ws=kaart&wom=Gemiddelde%20windsnelheid>
- Kenneth Reitz, R. P. (2011-2020). Opgehaald van The Hitchhiker's Guide to Python!:
<https://docs.python-guide.org/>
- Kripthor. (2014). *kripthor/teensy31-micloc Teensy 3.1 MicLoc project* . Opgehaald van github: <https://github.com/kripthor/teensy31-micloc>

- Kriphthor. (sd). *MicLoc*. Opgehaald van Rural Hacker:
<http://ruralhacker.blogspot.com/p/micloc.html>
- locatie-lokatie. (2011). Opgehaald van onzetaal: <https://onzetaal.nl/taaladvies/locatie-lokatie/>
- LOKKI, H. G. (2010). *SPATIALISATION IN AUDIO AUGMENTED REALITY USING FINGER SNAPS*. Aalto, FINLAND: Department of Media Technology, Aalto University.
- Maxim Integrated Products, Inc. (2012, December 06). *Low-Cost, Micropower, SC70/SOT23-8, Microphone Preamplifiers with Complete Shutdown*. Opgehaald van Maxim Intergrated Products:
<https://datasheets.maximintegrated.com/en/ds/MAX4465-MAX4469.pdf>
- Maximo Cobos, F. A. (2017, August).
A Survey of Sound Source Localization Methods in Wireless Acoustic Sensor Networks. Opgehaald van Researchgate:
https://www.researchgate.net/publication/319171966_A_Survey_of_Sound_Source_Localization_Methods_in_Wireless_Acoustic_Sensor_Networks
- MBI products. (2001). *Room acoustics*. Opgehaald van MBI products:
https://mbiproducs.com/static/mbi_1/media/misc/RoomAcoustics.pdf
- Onderzoeksmethoden. (2020, Februari 13). Opgehaald van Scribbr:
<https://www.scribbr.nl/category/onderzoeksmethoden/>
- Paul, S. (2009). *Binaural Recording Technology: A Historical Review and Possible Future Developments*. Acta Acustica.
- prjc. (2020). *Teensy USB development board*. Opgehaald van Electronics projects:
<https://www.pjrc.com/teensy/>
- prjc. (2020). *Teensyduino: Using the UART (real seial) with Teensy on the Arduino IDE*. Opgehaald van prjc: https://www.pjrc.com/teensy/td_uart.html
- Proakis, J. G., & Manolakis, D. G. (1996). *DIGITAL SIGNAL PROCESSING Principles, Algorithms, and Applications third edition*. PRENTICE-HALL INTERNATIONAL, INC.
- Python Software Foundation. (2020, June 07). *simple statements*. Opgehaald van Python 3.8.3 documentation: <https://docs.python.org/3/reference/expressions.html#yieldexpr>
- Raphael Wimmer, A. S. (2019). *On the Latency of USB-Connected Input Devices*.
- Reitz, K. (2020). *PEP8*. Opgehaald van PEP 8 — the Style Guide for Python Code:
<https://pep8.org/>
- Rose, B. (2020). *Comparing MEMS and Electret Condenser (ECM) Microphones*. Opgehaald van CUI Devices: <https://www.cuidevices.com/blog/comparing-mems-and-electret-condenser-microphones>
- Smith, S. W. (1997). Chapter 13: Continuous Signal Processing
<http://www.dspguide.com/ch13/2.htm>. In S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing* (p. Chapter 13). San Diego: California Technical Publishing . Opgehaald van <http://www.dspguide.com/ch13/2.htm>
- stackoverflow insights. (2020). *stackoverflow trends*. Opgehaald van stackoverflow insights:
https://insights.stackoverflow.com/trends?utm_source=s-owned&utm_medium=blog&utm_campaign=trends&utm_content=blog-link&tags=java%2Cpython
- voltage drop calculator. (2020). Opgehaald van calculator:
<https://www.calculator.net/voltage-drop-calculator.html>
- Wei Ma, X. L. (2019, May 14). *Phased microphone array for sound source localization with deep learning*. Aerospace systems. Opgehaald van
<https://link.springer.com/article/10.1007/s42401-019-00026-w>

WisFaq - de digitale vraagbaak voor wiskunde en wiskunde onderwijs. (sd). Opgehaald van WisFaq: <https://www.wisfaq.nl/bestanden/q9646img1.gif>
Zapponi, C. (2020). *github language stats - githut 2.0*. Opgehaald van githut 2.0: https://madnight.github.io/githut/#/pull_requests/2020/1

16 Bijlagen

Dit zijn alle bijlagen van het verslag die extra context geven aan het project/verslag, maar niet nodig zijn om het verslag te kunnen snappen.

16.1 Resultaten overige

De ruwe data van de testen die gedaan zijn met de locatie bepaling zullen in losse documenten geleverd worden. Elke folder heeft een naam met het format:

{jaar}-{maand}-{dag}-{uur}-{minuut}-{seconde} ({scenario nr.}) {x},{y}

Bijvoorbeeld: 2020-05-28--16-48-26(1) 1,0

Met x en y als de coördinaten waar ongeveer geklapt is bij het uitvoeren van de testen.

In elke folder zitten twee pdf bestanden en een .txt bestand.

De sounds.pdf file bevat grafieken van de geluiden die op elke microfoon zijn waargenomen en de cross-correlation tussen de verschillende microfoons, aangeduid door 'ftcorr' gevolgd door de "ID's" van de microfoons.

De location.pdf file bevat een heatmap van de punten waar de lokalisering methode het geluid plaatst. De schaal van de heatmap is logaritmisch en representeert de RMSE waarde van het berekende punt. Ook zijn hierop de microfoons en de meest waarschijnlijke voorspelling van de lokaliseringmethode geplot.

De data.txt file bevat een leesbare weergave van de data die verzameld is bij het verwerken van de geluidsdata.

16.2 Unittesten

Hier worden de unittesten die de desktop applicatie testen beschreven. Er wordt beschreven wat de functie is van elke test, waarom het getest wordt, hoe het getest wordt en of de test slaagt of niet.

Voor de SoundProcessingCoordinator wordt de handmatige kalibratie getest. Bij deze tests wordt de ingebouwde Python input functie gemockt. Hierbij worden een aantal vooraf bepaalde waarden meegegeven aan de kalibratie. De kalibratie berekend dan de locaties en hoeken van de microfoons. Er wordt dan getest of deze overeenkomen met wat er van tevoren berekend is. Er wordt een keer getest met waarden die resulteren in een rechte hoek, en een keer met twee gelijke lengtes.

De sound processing wordt ook getest. Hierin wordt getest of de frequenties van de signalen wel ongeveer goed berekend worden. Dit wordt gedaan door een sinusgolf met een bepaalde frequentie te genereren. Ook wordt er gekeken of de shift correct wordt berekend als twee golven dezelfde frequentie hebben. Er is ook een bad-weather test geschreven die test of de shifts NIET correct bepaald worden wanneer de frequenties van de geluiden van elkaar verschillen. In deze situatie klopt het dat de berekening van de shifts niet overeen komt, maar wanneer er in de praktijk shifts berekend worden, zullen er nooit perfecte sinusgolven gedetecteerd worden, en zijn de shifts makkelijker te herkennen.

Voor de locatiebepaling is er gekozen om een onorthodoxe manier van testen toe te passen. Omdat het er bij de locatiebepaling om gaat om zo dicht bij de echte waarde als mogelijk te

zitten is er gekozen om met random variabelen 100 keer te testen, en een acceptatiegrens stellen. Hierbij worden de geluidssnelheid, de afstand tussen de microfoons en de coördinaten van het te berekenen punt allemaal random bepaald. De TDOA's die bij de coördinaten horen van het te berekenen punt, worden bepaald met de helper functie `coords_to_tdoa()`. Voor SRP-PHAT is uiteindelijk de acceptatiegrens geworden dat 90% van de resultaten binnen 0.5 meter moeten zitten. Gemiddeld bereikt deze test tussen de 93% en 97% slagingspercentage. Voor de Triangulatie is de acceptatiegrens geworden dat 1% van de resultaten binnen 1 meter moeten zitten. De reden dat deze acceptatiegrens zo laag zit, is omdat het Triangulatie algoritme bagger is, en erg slechte resultaten oplevert. Dit komt er ook door omdat triangulatie drie punten oplevert, en er wordt getest of al deze drie punten binnen de acceptatiecriteria vallen.

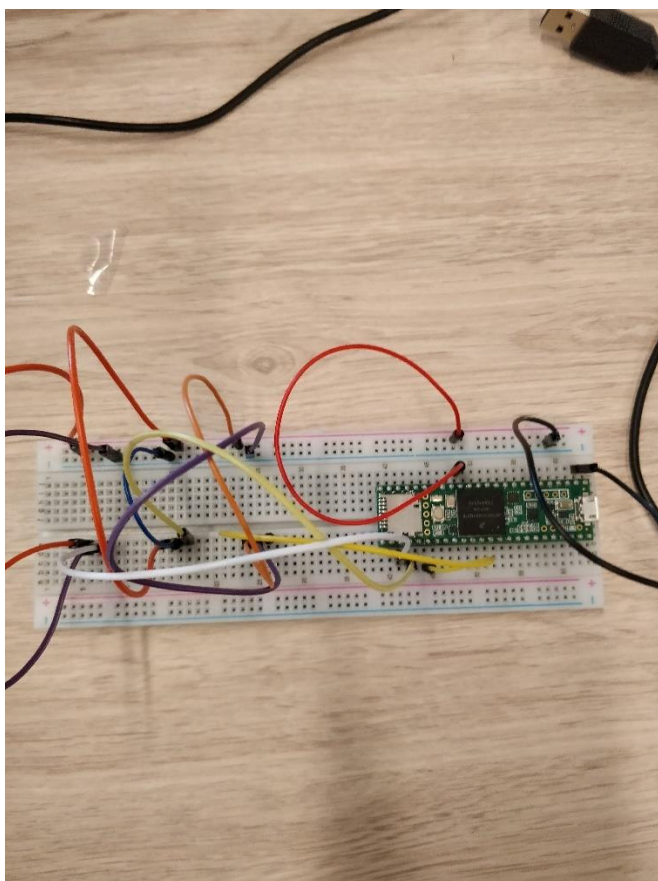
16.3 Foto's



Figuur 18 Test scenario 3



Figuur 19 Aansluiting XLR kabel en microfoon



Figuur 20 Breadboard met Teensy



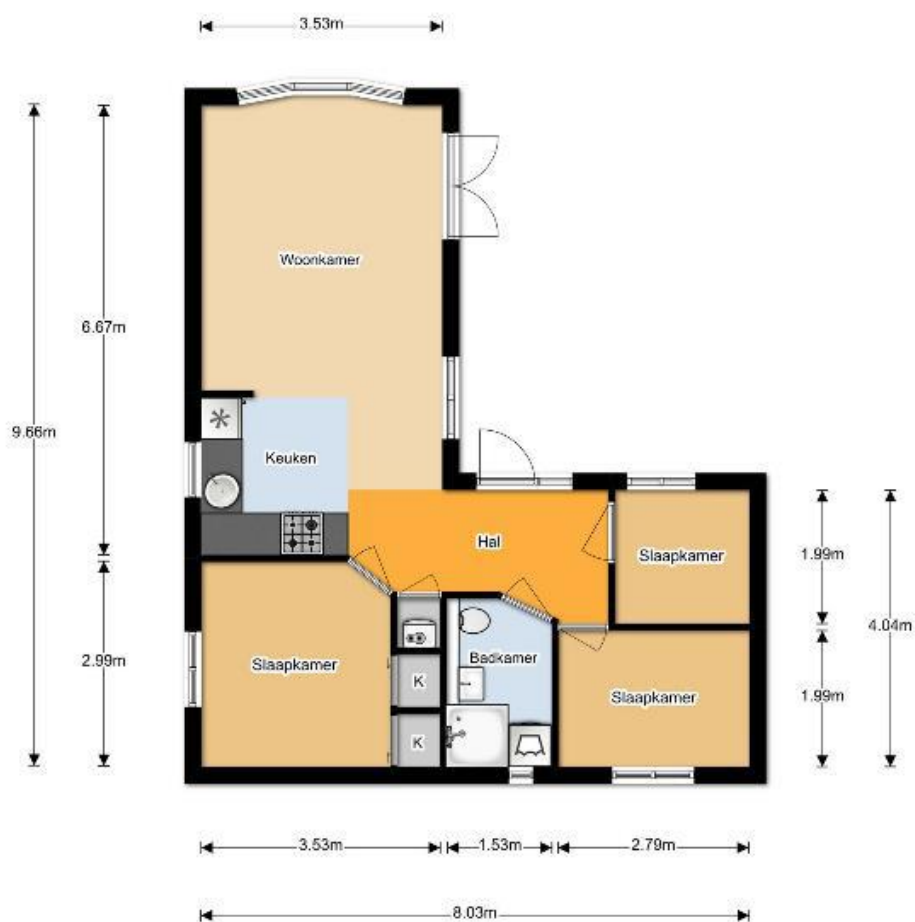
Figuur 21 Test scenario 2



Figuur 22 Test scenario 1, met bonus Tim



Figuur 23 Omgeving scenario 1 (Funda hoge bergweg 16, 2020)



Figuur 24 - Plattegrond scenario 1 (Funda hoge bergweg 16, 2020)

16.4 Reflectie

In dit hoofdstuk wordt gereflecteerd op alle gedeelten van het project. Hierin wordt er gereflecteerd op de verschillende fases van de afstudeerperiode.

Opstartfase

Tijdens de opstartfase had ik mijn twijfels over hoe de uitvoerperiode van mijn afstudeerstage er uit zou zien. Ik had voor mezelf nog niet goed genoeg duidelijk hoe de opdracht in elkaar zat in de ogen van de product-owner. Ondanks dit ben ik begonnen met een plan van aanpak en heb ik assumpties gedaan over hoe de product-owner het project voor zich zag. Hierdoor ben ik gedeeltelijk op een verkeerde voet begonnen, en heb ik me ingesteld op mijn visie van het project. Hierdoor kwam ik aan het begin van het project een paar dagen achter op schema te liggen, al nadat het plan van aanpak ingeleverd was. Als er eerder duidelijke requirements door waren gesproken met de product-owner, in plaats van dat er assumpties gedaan werden, was er vanaf het begin een duidelijker beeld van de opdracht en zou er tijd bespaard zijn.

Ik ben wel tevreden over de snelheid en de kwaliteit van mijn plan van aanpak. Daan, mede afstudeerder, heeft officiële Microsoft certificering voor Word, en heeft mij geholpen met een aantal handige trucjes om documentatie goed op orde te kunnen krijgen. Dit heb ik ook gebruikt voor dit verslag.

Onderzoeksfase

De motivatie om aan het project te werken was er. Ik wou graag wat doen met hardware, aangezien ik hier erg geïnteresseerd in ben. Een obstakel hiervan was dat ik alle kennis die ik nodig zou hebben over hardware tijdens de afstudeerperiode zou moeten vergaren. Aan het begin van het project wist ik niet goed waar ik moest beginnen met de hardware kant. Wat heel erg hielp bij deze beginproblemen was de hulp van mijn collega's. De vele behulpzame afstudeerders en collega's zorgden er voor dat ik nooit te lang op 1 probleem vast kwam te zitten. Ze hebben er voor gezorgd niks voor me voor te kauwen, maar hebben me vaak een duwtje in de goede richting gegeven met wat feedback of advies. Hiervan heb ik enorm veel geleerd, en daar ben ik ze enorm dankbaar voor.

Tijdens de onderzoeksfase ben ik ook een aantal lokalisering methoden uit gaan proberen. Dit bleek achteraf extreem nuttig te zijn, aangezien het bleek dat de triangulatie methode een stuk onvoorspelbaarder en minder accuraat bleek te zijn dan van tevoren gedacht. Hierdoor is tijdens de onderzoeksfase meer gekeken naar SRP-PHAT en hoe dat een verbetering is tegenover triangulatie.

Implementatiefase

Op het moment dat ik begon aan de implementatiefase kwam Covid-19 om de hoek kijken. Dit was tijdens de afstudeerperiode verreweg het grootste obstakel. Ik ben een persoon die goed kan werken in een omgeving met collega's en sparringpartners. Wanneer ik opeens 40 uur in de week thuis moest werken, ging de voortgang die ik boekte heel snel naar beneden. Zelf had ik gedacht dat het niet zo erg zou zijn als dat het was, maar het had een hele grote impact op mijn vermogen om efficiënt te kunnen werken. Om van kwaad tot erger te gaan, overleed er plotseling een goede vriend van me. Om dit allemaal te verwerken heb ik het afstuderen tijdelijk op een lager pitje gezet.

Wanneer ik merkte, dat als het zo doorging ik het niet zou gaan redden, heb ik aan de bel getrokken. Ik heb gevraagd bij ALTEN of ik mogelijk op kantoor zou kunnen werken. Er kwam uiteindelijk een schema waarin ik me kon inschrijven om op kantoor te werken. Aan het begin heb ik dat 1 dag in de week gedaan. Merkend dat dat ook niet genoeg is ben ik naar drie dagen in de week op kantoor werken gegaan. Hierdoor kon ik een goede balans vinden tussen thuis werken in een werkomgeving zitten. Ook maakte het op kantoor werken het makkelijker om mijn implementatie te testen, aangezien er daar veel meer ruimte is dan dat ik thuis beschikbaar heb.

Ik kwam er in deze periode wel achter dat het implementeren van de hardware een stuk makkelijker was dan van tevoren gedacht. Er kwamen een hele hoop extra kleine aandachtspuntjes bij, zoals stabiliteit, debuggen en het meeslepen van materialen, maar dat was allemaal erg te overzien.

Wel denk ik dat ik me iets te veel heb gefocust op de hardware en natuurkundige aspecten van het project. Daarom is de software kant een stuk krapper geworden en heb ik de implementatie niet zo uitgebreid gedaan als dat ik uiteindelijk had gewild. Dit had geprioriteerd moeten worden, aangezien ik aan het afstuderen ben voor een Software engineering opleiding.

Test en afrondingsfase

Ik ben te laat begonnen aan de testfase. Dit komt onder andere door alle omstandigheden die zich voor hebben gedaan. Hierdoor zijn de tests die zijn uitgevoerd niet zo uitgebreid als dat ik had gehoopt. Er hadden nog een hele hoop andere tests, met meer opstellingen, andere instellingen en in compleet andere omgevingen gedaan kunnen worden als er meer tijd was.

Van een aantal hele behulpzame collega's heb ik nog wat handige tips gekregen voor tijdens het testen, waar ik ze erg dankbaar voor ben. Bijvoorbeeld dat het niet echt nut heeft om voor een proof of concept van dit project ook andere geluiden proberen te gebruiken voor de locatiebepalingen, omdat dat niet relevant is.

Qua documentatie heb ik het te veel uitgesteld tot het einde. Hierdoor heb ik vooral het gedeelte resultaten verwerking en software documentatie niet op tijd af gehad voor de concept-inlevering.

Dit heeft er voor gezorgd dat ik minder feedback op deze onderdelen heb kunnen krijgen. Ook heeft dit ervoor gezorgd dat ik nog veel te veel werk in de laatste twee weken moest doen. Dit resulteerde in vele avonden en nachten extra werk aan het verslag.

Dit had allemaal voorkomen kunnen worden als ik eerder was begonnen en er op tijd voor gezorgd had dat het onderzoek gedeelte klaar was.