



Faculteit Ingenieurswetenschappen en Architectuur

Vakgroep Informatietechnologie

Voorzitter: Prof. Dr. Ir. Daniël De Zutter

Realtime signaal synchronisatie met acoustic fingerprinting

door

Ward Van Assche

Promotoren: Dr. Marleen Denert, Joren Six

Scriptiebegeleider: Prof. Helga Naessens

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: informatica

Academiejaar 2015–2016

Voorwoord

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ac tristique risus. Morbi sit amet porta ex. Vivamus at blandit mi, eget finibus nisi. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla vehicula efficitur rutrum. Quisque ac condimentum ligula, at tempus leo. Sed quis felis erat. Proin volutpat et odio ac finibus. Cras eu ultrices augue. Aliquam finibus lacus ut erat placerat, non ullamcorper arcu porttitor. Phasellus nisi metus, porttitor pretium diam ac, euismod ultricies nunc. Proin auctor pulvinar tellus eu egestas. Fusce non nisl commodo, luctus diam a, interdum ex. Pellentesque quis enim sit amet felis suscipit suscipit sed ac nunc. Integer porta, ipsum non placerat tincidunt, est quam eleifend urna, sed aliquam libero felis ac odio.

Proin non maximus est. Nullam eu magna et mauris faucibus dictum. Vivamus vitae commodo enim, dignissim consequat magna. Vivamus et elementum velit. Suspendisse justo lacus, euismod in nibh nec, dictum eleifend libero. Vivamus dapibus dignissim nibh, eu luctus nunc. Vivamus hendrerit massa in velit cursus pharetra. Aliquam consectetur dapibus mi at semper.

Nam rhoncus lectus risus. Vivamus dolor justo, viverra nec libero eget, porttitor rhoncus sapien. Proin commodo erat leo, eget dignissim massa consequat non. Sed pharetra eget libero nec faucibus.

Toelating tot bruikleen

“De auteur geeft de toelating deze scriptie voor consultatie beschikbaar te stellen en delen van de scriptie te kopiëren voor persoonlijk gebruik.

Elk ander gebruik valt onder de beperkingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze scriptie.”

Ward Van Assche, juni 2016

Realtime signaal synchronisatie met acoustic fingerprinting

door

Ward Van Assche

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: informatica

Academiejaar 2015–2016

Promotoren: Dr. Marleen Denert, Joren Six
Scriptiebegeleider: Prof. Helga Naessens

Faculteit Ingenieurswetenschappen en Architectuur
Universiteit Gent

Vakgroep Informatietechnologie
Voorzitter: Prof. Dr. Ir. Daniël De Zutter

Samenvatting

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ac tristique risus. Morbi sit amet porta ex. Vivamus at blandit mi, eget finibus nisi. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla vehicula efficitur rutrum. Quisque ac condimentum ligula, at tempus leo. Sed quis felis erat. Proin volutpat et odio ac finibus. Cras eu ultrices augue. Aliquam finibus lacus ut erat placerat, non ullamcorper arcu porttitor. Phasellus nisi metus, porttitor pretium diam ac, euismod ultricies nunc. Proin auctor pulvinar tellus eu egestas.

Trefwoorden

synchronisatie, realtime, datastromen, musicologie, acoustic fingerprinting

Realtime signal synchronization with acoustic fingerprinting

Ward Van Assche

Supervisor(s): Joren Six, Marleen Denert

Abstract—Sed nec tortor in libero rutrum pellentesque et gravida turpis. Phasellus gravida neque vitae elit fringilla, a efficitur purus sollicitudin. Proin lacus est, suscipit sed nibh ac, hendrerit eleifend leo. Suspendisse quis semper leo. Duis non elit commodo, sodales ex non, venenatis diam. Sed libero tortor, hendrerit et sollicitudin ut, facilisis vitae odio. Fusce vitae mi odio.

Keywords—kernwoord1, kernwoord2, kernwoord 3, kernwoord 4

I. INTRODUCTION

SED Sed nec tortor in libero rutrum pellentesque et gravida turpis. Phasellus gravida neque vitae elit fringilla, a efficitur purus sollicitudin. Proin lacus est, suscipit sed nibh ac, hendrerit eleifend leo. Suspendisse quis semper leo. Duis non elit commodo, sodales ex non, venenatis diam. Sed libero tortor, hendrerit et sollicitudin ut, facilisis vitae odio. Fusce vitae mi odio. Cras vitae quam bibendum, elementum velit ut, varius enim. Donec sagittis elit ligula, laoreet viverra felis rhoncus nec. Donec mattis metus pretium, pulvinar enim a, luctus nunc. Pellentesque quis suscipit leo.

II. SECTIE

A. Subsectie

Sed nec tortor in libero rutrum pellentesque[1] et gravida turpis. Phasellus gravida neque vitae elit fringilla, a efficitur purus sollicitudin. Proin lacus est, suscipit sed nibh ac, hendrerit eleifend leo. Suspendisse quis semper leo. Duis non elit commodo, sodales ex non, venenatis diam. Sed libero tortor, hendrerit et sollicitudin ut, facilisis vitae odio. Fusce vitae mi odio. Cras vitae quam bibendum, elementum velit ut, varius enim. Donec sagittis elit ligula, laoreet viverra felis rhoncus nec. Donec mattis metus pretium, pulvinar enim a, luctus nunc. Pellentesque quis suscipit leo.

B. Andere subsectie

Sed nec tortor in libero rutrum pellentesque et gravida turpis. Phasellus gravida neque vitae elit fringilla, a efficitur purus sollicitudin. Proin lacus est, suscipit sed nibh ac, hendrerit eleifend leo. Suspendisse quis semper leo. Duis non elit commodo, sodales ex non, venenatis diam. Sed libero tortor, hendrerit et sollicitudin ut, facilisis vitae odio. Fusce vitae mi odio. Cras vitae quam bibendum, elementum velit ut, varius enim. Donec sagittis elit ligula, laoreet viverra felis rhoncus nec. Donec mattis metus pretium, pulvinar enim a, luctus nunc. Pellentesque quis suscipit leo.

Sed nec tortor in libero rutrum pellentesque et gravida turpis. Phasellus gravida neque vitae elit fringilla, a efficitur purus sollicitudin. Proin lacus est, suscipit sed nibh ac, hendrerit eleifend

leo. Suspendisse quis semper leo. Duis non elit commodo, sodales ex non, venenatis diam. Sed libero tortor, hendrerit et sollicitudin ut, facilisis vitae odio. Fusce vitae mi odio. Cras vitae quam bibendum, elementum velit ut, varius enim. Donec sagittis elit ligula, laoreet viverra felis rhoncus nec. Donec mattis metus pretium, *pulvinar* enim a, luctus nunc. Pellentesque quis suscipit leo.

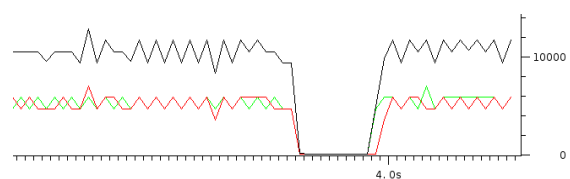


Fig. 1. Detailed capture of the stream at the moment of a handover between two simulated AP's with a strong signal. Notice the gap of 100 ms.

III. SECTIE

Aenean auctor congue nisi, volutpat porta urna lobortis in. Donec accumsan fermentum lectus, sed aliquet lectus gravida eget. Ut turpis quam, fermentum eget sem sed, euismod facilisis sem. Etiam a sollicitudin purus. Vestibulum quis nisl et nibh condimentum suscipit tristique eget quam. Aenean eget varius lectus. Maecenas sit amet mi augue. Nullam semper ex et facilisis ullamcorper. Cras volutpat ornare arcu, ac suscipit nisi pellentesque iaculis. Vivamus sit amet ipsum consequat, egestas massa varius, aliquam lorem. Aenean ornare iaculis dolor, eget efficitur elit. Maecenas ut massa ac tortor hendrerit pulvinar a in ante.

IV. CONCLUSION

The simulation results show a nice advantage for the moving cell[3] concept. The traditional handover problem can be avoided so a workable model for broadband access on trains seems realistic. But there needs to be done a lot of research to make RAU's and RoF as reliable and cheap as possible.

REFERENCES

- [1] Joren Six, Olmo Cornelis, and Marc Leman, "TarsosDSP, a Real-Time Audio Processing Framework in Java," in *Proceedings of the 53rd AES Conference (AES 53rd)*. 2014, The Audio Engineering Society.
- [2] Joren Six and Marc Leman, "Panako - A Scalable Acoustic Fingerprinting System Handling Time-Scale and Pitch Modification," in *Proceedings of the 15th ISMIR Conference (ISMIR 2014)*, 2014.
- [3] Joren Six and Marc Leman, "Synchronizing Multimodal Recordings Using Audio-To-Audio Alignment," *Journal of Multimodal User Interfaces*, vol. 9, no. 3, pp. 223–229, 2015.
- [4] A. L. Wang, "An industrial-strength audio search algorithm," in *ISMIR 2003, 4th Symposium Conference on Music Information Retrieval*, 2003, pp. 7–13.

Inhoudsopgave

Extended abstract	4
Gebruikte afkortingen	iii
1 Inleiding	1
1.1 Probleemschets	1
1.2 Doel van het onderzoek	3
1.3 Huidige implementatie	5
1.3.1 TarsosDSP	5
1.3.2 Panako	6
2 Applicatie-ontwerp	7
3 Accoustic fingerprinting	8
3.1 Optimalisaties	8
3.2 Parameterisatie	8
3.3 Uitgevoerde testen	8
4 Kruiscovariantie	9
4.1 Optimalisaties	9
4.2 Parameterisatie	9
4.3 Uitgevoerde testen	9

5 Implementatie in MAX/MSP	10
6 Conclusie	11
Referentielijst	12
Lijst van figuren	13
Lijst van figuren	13
Lijst van tabellen	14
Lijst van tabellen	14
Lijst van codefragmenten	15
Lijst van codefragmenten	15

Gebruikte afkortingen

IPEM Instituut voor Psychoakoestiek en Elektronische Muziek

Hoofdstuk 1

Inleiding

1.1 Probleemschets

Het probleem dat in deze masterproef zal worden onderzocht doet zich heel specifiek voor bij verschillende experimenten die aan het IPeM worden uitgevoerd. Het IPeM is de onderzoeksinstelling van het departement musicologie aan Universiteit Gent. De focus ligt vooral op onderzoek naar de interactie van muziek op fysieke aspecten van de mens zoals dansen, sporten en fysieke revalidatie.[1]

Om de relatie tussen muziek en beweging te kunnen onderzoeken worden er allerhande sensoren gebruikt die bepaalde gebeurtenissen omzetten in analyseerbare data. Een mogelijk experiment (puur imaginair) kan bijvoorbeeld gebruik maken van een videocamera om een persoon te filmen en verschillende accelerometers om de bewegingen van de persoon te detecteren. Er wordt ook muziek afgespeeld zodat men met behulp van de videobeelden en de data van de accelerometers kan analyseren hoe de persoon reageert op de afgespeelde muziek.

Om het zojuist beschreven experiment verder te onderzoeken moeten er minstens drie datastromen worden geanalyseerd: de videobeelden, de data van de accelerometer(s), en

de afgespeelde audio. Een probleem dat zich hierbij voordoet is de synchronisatie van deze verschillende datastromen. Om een goede analyse mogelijk te maken is het zeer gewenst dat men exact weet (tot op de milliseconde nauwkeurig) wanneer een bepaalde gebeurtenis in een datastroom zich heeft voorgedaan, zodat men deze gebeurtenis kan vergelijken met de gebeurtenissen in de andere datastromen.

Bij het IPEM maakt men gebruik van een systeem dat gebruikt maakt van audio opnames om de datastromen te kunnen synchroniseren. Het principe werkt als volgt: men zorgt ervoor dat elke datastroom vergezeld wordt met een perfect gesynchroniseerde audiostroom, afkomstig van een opname van het omgevingsgeluid. In het voorgaande experiment is dit eenvoudig te verwezenlijken. Bij de videobeelden kan automatisch een audiospoor mee worden opgenomen. De accelerometer kan geplaatst worden op een microcontroller (bijvoorbeeld een Arduino), hierop kan een klein microfoontje geplaatst worden. Aangezien beide componenten zo dicht op de hardware geplaatst zijn is de latency tussen beide datastromen te verwaarlozen. De afgespeelde audio kan gebruikt worden als referentie, aangezien dit uiteraard al een perfecte weergave is van het omgevingsgeluid. Na het uitvoeren van het experiment beschikt men dus over drie datastromen, waarbij elke datastroom vergezeld is van een quasi perfect synchrone opname van het omgevingsgeluid (dat in de ruimte waar het experiment is uitgevoerd voor elke opname gelijk is). Het probleem van de synchronisatie van de verschillende datastromen kan bijgevolg gereduceerd worden tot het synchroniseren van de verschillende audiostromen.

Door de typisch eigenschappen van geluid is het helemaal niet zo moeilijk om verschillende audiostromen te synchroniseren. Bij het IPEM heeft men een bepaald systeem ontwikkeld dat dergelijke synchronisatie mogelijk maakt met behulp van *accoustic fingerprinting*. Accoustic fingerprinting is vooral bekend van de enorm populaire smartphone app voor muziek identificatie: *Shazam*. Hierbij wordt dit principe gebruikt om een kort stukje opge-

nomen audio te vergelijken met een gigantische database van akoestische fingerprints.¹ Na het uitvoeren van het fingerprinting algoritme is het mogelijk om een bijkomend algoritme uit te voeren namelijk: synchronisatie met *kruiscovariantie*. Dit algoritme zorgt er voor dat de synchronisatie een veel nauwkeuriger resultaat oplevert.

Ondanks het feit dat er al een systeem bestaat om datastromen te synchroniseren zijn er in de praktijk toch nog heel wat beperkingen. De grootste beperking is dat het synchronisatieproces pas kan worden uitgevoerd wanneer het experiment is afgelopen, en dit volledig handmatig. De opgenomen audiobestanden moet worden verzameld op een computer, vervolgens kan met behulp van de audiobestanden de latency worden berekend tussen elke datastroom worden berekend. Vervolgens kunnen de datastromen worden gesynchroniseerd. Voor de musicologen die deze experimenten uitvoeren is deze werkwijze veel te omslachtig. Daarom is een eenvoudigere realtime systeem om de synchronisatie uit te voeren zeer gewenst.

Een ander probleem is iets vager en minder duidelijk te omschrijven. De resultaten van het kruiscovariantie algoritme bevatten soms afwijkingen die moeilijk te verklaren zijn. De precieze oorzaak hiervan, en hoe dit kan worden opgelost zal ook worden onderzocht. Ook is het kruiscovariantie algoritme in vergelijking met het accoustic fingerprinting algoritme véél gevoeliger voor storingen en ruis, veroorzaakt door slechte opnames. Aangezien de opnameapparatuur (zeker op microcontrollers) bij de uit te voeren experimenten vaak van slechte kwaliteit is, is het belangrijk om de algoritmes robuust genoeg te maken zodat ze hier niet over struikelen.

1.2 Doel van het onderzoek

Dit onderzoek wil drie zaken bereiken:

¹ In deze scriptie wordt er verondersteld dat het principe achter accoustic fingerprinting gekend is bij de lezer. Een grondige theoretische bespreking van dit algoritme is te vinden in de paper van Wang, één van de oprichters van Shazam ltd. De implementatie die bij het IPFM gebruikt wordt voor de synchronisatie van de audiostromen wordt verderop in deze scriptie besproken.

Optimalisatie van algoritmes

Het testen en bug-vrij maken van de synchronisatiealgoritmes. Ook moeten ze eventueel worden aangepast zodat ze in een realtime implementatie gebruikt kunnen worden. Het beoogde doel is dat de algoritmes in staat moeten kunnen zijn om audio opgenomen met een basic microfoon op een microcontroller te synchroniseren met een nauwkeurigheid van minstens 1 milliseconde.

Implementatie van een Java bibliotheek

Het schrijven van een bibliotheek in Java die gebruik maakt van deze algoritmes om audiostromen te synchroniseren. Deze bibliotheek moet kunnen worden aangeroepen vanuit eender welke andere Java applicatie, en moet periodiek de huidige latency per audiostroom teruggeven.

Implementatie van een MAX/MSP module

De implementatie van een module in MAX/MSP² die realtime synchronisatie mogelijk moet maken via een interface die bruikbaar is voor musicologen/onderzoekers met een beperkte informatica achtergrond.

Deze module moet in staat zijn om verschillende datastromen als input binnen te krijgen, de synchronisatiebibliotheek aan te roepen, en de gesynchroniseerde datastromen als output terug te geven. Een andere Max module kan er dan voor zorgen dat deze data wordt weggeschreven naar een WAVE-bestand.

² Cycling '74 Max/MSP is een softwarepakket en een visuele programmeertaal waarmee audio, video en multimedia kan worden verwerkt met behulp van onafhankelijke modules. Deze modules kunnen met elkaar worden om zo complexe zaken te bereiken. Buiten de standaard meegeleverde modules is het ook mogelijk om zelf modules te schrijven.[2]

1.3 Huidige implementatie

Om het vervolg van deze scriptie goed te kunnen begrijpen is een introductie tot de bestaande toepassingen waarop wordt verder gebouwd, noodzakelijk. De twee belangrijkste bibliotheken zijn TarsosDSP en Panako.

1.3.1 TarsosDSP

TarsosDSP³ is een Java bibliotheek voor realtime audio analyse en verwerking ontwikkeld aan het IPEM door Six, Cornelis, and Leman. De bibliotheek bevat een groot aantal algoritmes voor audioverwerking en kan nog verder worden uitgebreid.

TarsosDSP werkt met een zeer eenvoudige *processing pipeline*. We kunnen zelf een processing pipeline creëren door een instantie aan te maken van de klasse `AudioDispatcher`. Het aanmaken van een `AudioDispatcher` is het eenvoudigst met behulp van de klasse `AudioDispatcherFactory`. Deze Factory klasse voorziet in statische methodes om een `AudioDispatcher` aan te maken met als input een audiobestand, een array van `float` waarden, of de data van een microfoon. Aan deze pipeline kunnen vervolgens verschillende `AudioProcessors` worden toegevoegd. Een `AudioProcessors` moet verplicht de `process` en `processingFinished` methodes implementeren. De `process` methode heeft als parameter een object van de klasse `AudioEvent`. Dit object bevat een audio blok, voorgesteld als `float` array met waarden tussen -1.0 en 1.0. De grootte van dit blokje audio, en de mate van overlapping tussen de opeenvolgende blokjes audio is instelbaar. Verder bevat het `AudioEvent` object ook nog andere metadata zoals onder meer een *timestamp*.

Afhankelijk van de implementatie van de `process` methode kan de audiostroom op een bepaalde manier verwerkt, geanalyseerd of gewijzigd worden.

³ Deze bibliotheek is veel uitgebreider dan wat hier wordt geschreven. Deze paper geeft hierover meer informatie: http://0110.be/files/attachments/415/aes53_tarsos_dsp.pdf[4]

1.3.2 Panako

Panako⁴ is ook een Java bibliotheek, samen met enkele applicaties die deze bibliotheek aanroepen. Panako is ook ontwikkeld aan het IPEM door Six and Leman en maakt gebruik van TarsosDSP als onderliggende technologie voor de verwerking van de audio.

Panako bevat een open-source implementatie van het accoustic fingerprinting algoritme beschreven in de paper van Wang. Dit algoritme is verder uitgebreid zodat audio waarbij de toonhoogte verhoogd of verlaagd is, of de audio sneller of trager is afgespeeld toch gedetecteerd kan worden.

De bibliotheek bevat verschillende applicaties die gebruik maken van dit algoritme. Zo is het mogelijk om de fingerprints van een geluidsfragment te bekijken, matches tussen verschillende geluidsfragmenten te visualiseren, en grafisch te experimenteren met de verschillende parameters.

Er is ook een applicatie beschikbaar om verschillende geluidsfragmenten te synchroniseren. Deze applicatie maakt buiten van het accoustic fingerprinting algoritme ook nog gebruik van het kruiscovariantie algoritme, dit is ook geïmplementeerd in Panako. Op welk principe dat dit algoritme gebaseerd is, en hoe dit geïmplementeerd is wordt verder in deze scriptie besproken. Wanneer de latency tussen de verschillende fragmenten is gedetecteerd genereert de applicatie een shell script dat met behulp van FFmpeg⁵ stukjes van de geluidsbestanden wegnipt of stilte toevoegt. Het resultaat is dat na het uitvoeren van het script de geluidsbestanden gesynchroniseerd zijn.

⁴ Ook deze bibliotheek wordt meer uitvoerig besproken in een paper geschreven door de auteurs: http://0110.be/files/attachments/415/ismir_2014_panako_fingerprinter.pdf[3] ⁵ Dit is een command-line applicatie voor het opnemen, verwerken en streamen van audio en video. Meer informatie: <https://ffmpeg.org/>

Hoofdstuk 2

Applicatie-ontwerp

Hoofdstuk 3

Acoustic fingerprinting

3.1 Optimalisaties

3.2 Parameterisatie

3.3 Uitgevoerde testen

Hoofdstuk 4

Kruiscovariantie

4.1 Optimalisaties

4.2 Parameterisatie

4.3 Uitgevoerde testen

Hoofdstuk 5

Implementatie in MAX/MSP

Hoofdstuk 6

Conclusie

Referentielijst

- [1] Ipem - systematic musicology. <https://www.ugent.be/lw/kunstwetenschappen/en/research-groups/musicology/ipem>. [Online; geraadpleegd 05-maart-2016].
- [2] Cycling '74 max. <https://cycling74.com/>. [Online; geraadpleegd 05-maart-2016].
- [3] Joren Six and Marc Leman. Panako - A Scalable Acoustic Fingerprinting System Handling Time-Scale and Pitch Modification. In *Proceedings of the 15th ISMIR Conference (ISMIR 2014)*, 2014.
- [4] Joren Six, Olmo Cornelis, and Marc Leman. TarsosDSP, a Real-Time Audio Processing Framework in Java. In *Proceedings of the 53rd AES Conference (AES 53rd)*. The Audio Engineering Society, 2014.
- [5] Avery Li-Chun Wang. An industrial-strength audio search algorithm. In *ISMIR 2003, 4th Symposium Conference on Music Information Retrieval*, pages 7–13, 2003.

Lijst van figuren

Lijst van tabellen

Lijst van codefragmenten