Masterproef

Realtime signaalsynchronisatie met accoustic fingerprinting

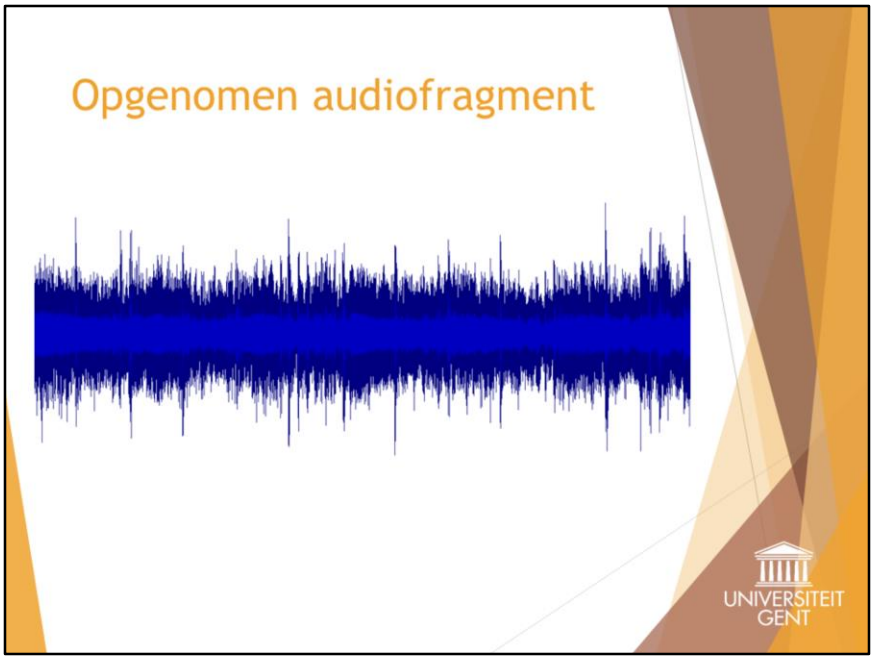
Ward Van Assche
28 april 2016

Onderzoeksgroep: IPEM

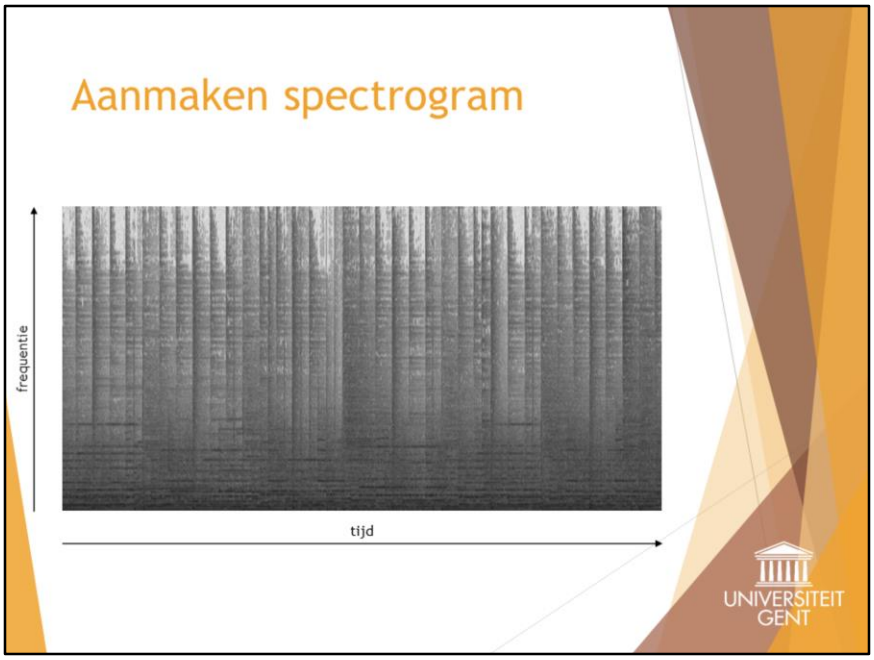
UNIVERSITEIT GENT

Wie kent er deze app? Iedereen waarschijnlijk?
En wie weet er hoe deze app er in slaagt om zo snel deze liedjes te detecteren?

Dit gebeurd met accoustic fingerprinting.
Bij accoustic fingerprinting wordt er op basis van een audiofragment (de opname) een verzameling fingerprints berekent. Die verzameling fingerprints kan gezien worden als de DNA van het liedje. Het zoeken naar fingerprints gebeurd op basis van de spectrale pieken. Dit zijn pieken het spectrogram van de audio. Dit klinkt vrij complex maar dit is het eigenlijk niet. Dit zal duidelijker worden na een voorbeeldje.

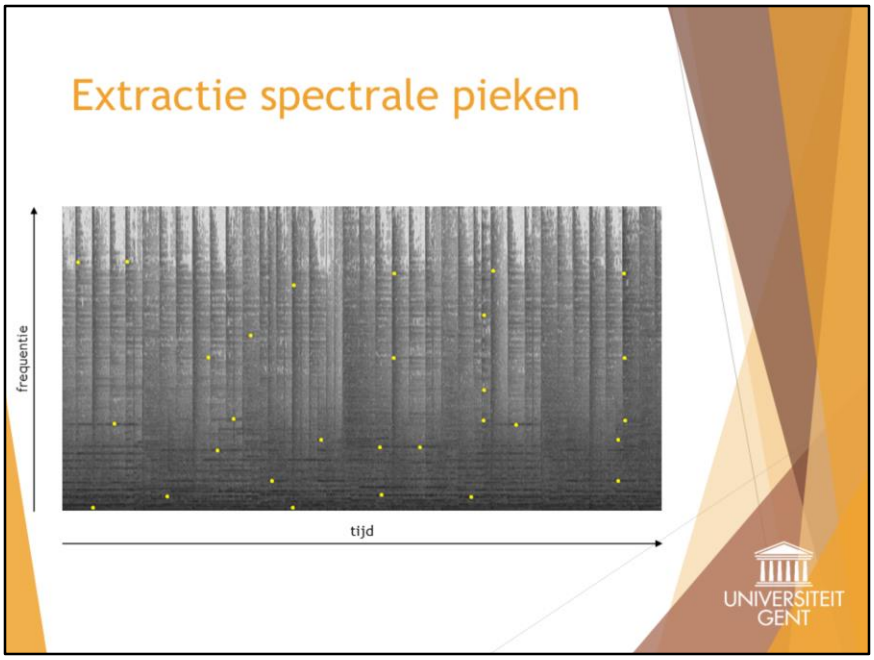


Stel je zit op café en hoort een liedje waarvan je kost wat het kost de titel en band van wil te weten komen. Je pakt je smartphone en laat Shazam er op los. Er wordt een opname gemaakt van pakweg 10 seconden. Dit fragmentje wordt doorgestuurd naar de Shazam servers en daar begint het echte werk: het accoustic fingerprinting.

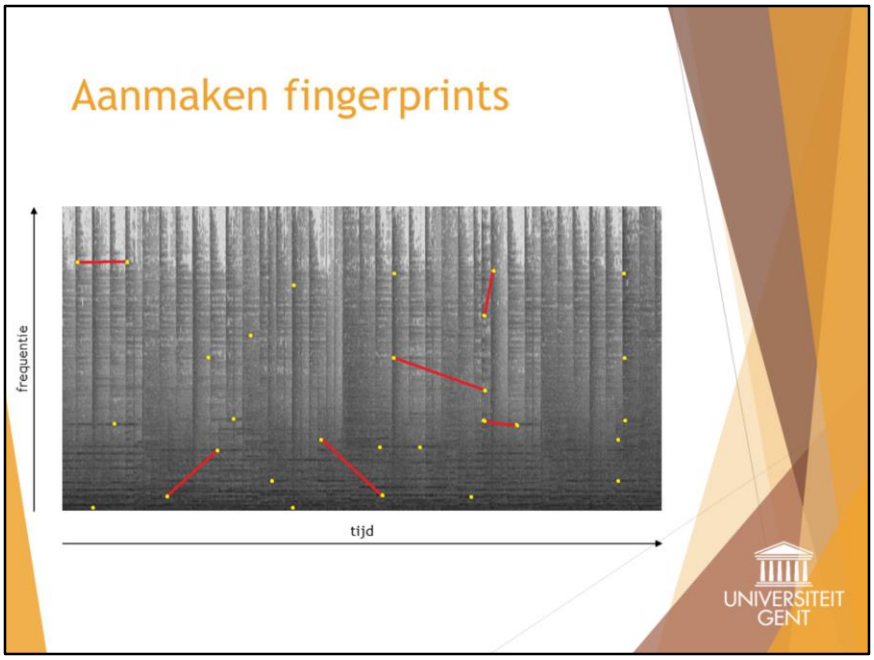


Eerst en vooral wordt er van het audiofragment een spectrogram aangemaakt. Dit is een soort van grafiek waarbij de frequentie is uitgezet ten opzichte van de tijd. Hoe donker een bepaald vlekje is, met hoe meer energie deze frequentie op dat moment voorkomt in de muziek.
Veel eigenschappen van de muziek zijn zichtbaar in het spectrogram. Uit de verticale lijnen kan je het ritme afleiden, de horizontale lijnen worden bepaald door de melodie.

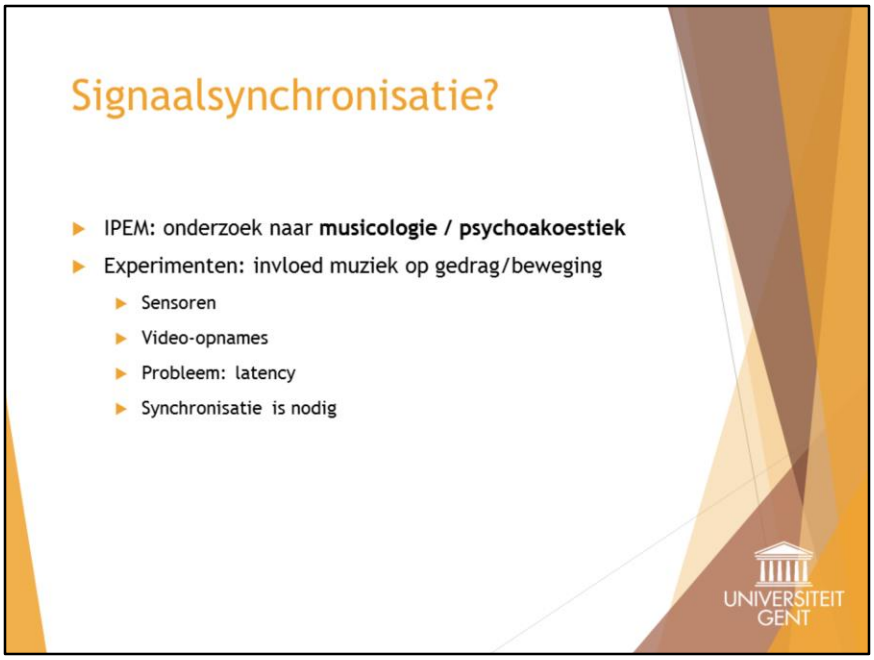
Daarstraks hebben ik gezegd dat fingerprints worden opgebouwd met behulp van spectrale pieken. Dit zijn plaatsen in het spectrogram waar de energie hoger is dan alle nabijgelegen plaatsen. In het spectrogram van onze opname heb ikzelf ook handmatig enkele spectrale pieken aangeduid.



Nadat de spectrale pieken bepaald zijn kunnen de fingerprints worden aangemaakt. Een fingerprint is een verbinding tussen twee spectrale pieken. Hoe er precies beslist wordt welke spectrale pieken verbonden worden hangt af van heel wat parameters, daar ga ik niet dieper op ingaan.



Nadat de fingerprints geconstrueerd zijn kan er gezocht worden naar welk liedje we nu eigenlijk hebben opgenomen. Hiervoor heeft shazam een enorme database met daarin de fingerprints van miljoenen liedjes. De fingerprints van het opgenomen geluidsfragment worden vergeleken met de fingerprints uit deze database, en zo probeert shazam te achterhalen welk liedje we hebben opgenomen. Elke fingerprint kan héél compact, als een getal, worden opgeslaan. Dit getal wordt een hash genoemd. Aangezien het vergelijken van getallen door computers heel snel kan gebeuren kan shazam zo vlug resultaten vinden.



Allemaal heel interessant, maar wat heeft dit nu met mijn masterpoef te maken? Wel, ik doe mijn masterproef bij het IPEM, dit is een onderzoeksinstelling aan universiteit gent, die onderzoek doet naar alles wat met muziek en audio te maken heeft. Er worden veel experimenten uitgevoerd waarbij er gekeken wordt naar wat de invloed is van muziek op gedrag of beweging van de mens. Om het gedrag of de beweging van personen in zo'n experiment te analyseren wordt er gebruik gemaakt van allerhande sensoren en video-opnames. Na zo'n experiment beschikken de onderzoekers over grote hoeveelheden data. Het grote probleem is dat elke sensor of opname een bepaalde vertraging of latency heeft. Voordat de gegevens door de onderzoekers geanalyseerd kunnen worden moet er voor worden gezorgt dat ze synchroon zijn.

Bij het IPEM heeft men een synchronisatie systeem ontwikkeld dat gebruik maakt van audio. Daarom wordt er aan elke sensor een microfoon gekoppeld die het omgevingsgeluid opneemt. Ook wordt elke video-opname voorzien van een audiospoor. Door er voor te zorgen dat de opname zeer dicht bij de sensor gebeurd kan men er vanuitgaan dat de latency van de geluidsopname hetzelfde is als de latency van de datastream van de sensor.
Het probleem van het synchroniseren van de datastreams hebben we eigenlijk kunnen omzetten naar het synchroniseren van opnames van het omgevingsgeluid. Aangezien het omgevingsgeluid van elke sensor hetzelfde is: de afgespeelde muziek is het probleem veel eenvoudiger op te lossen.

Heel toevallig bestaat er een methode waarmee audiofragmenten met elkaar kunnen worden vergeleken: accoustic fingerprinting! In tegenstelling tot de manier waarop Shazam het algoritme gebruikt gaan we de opnames niet matchen met een gigantische database maar worden de opnames van de verschillende sensoren met elkaar gematcht. Dit heb ik daarstraks niet vermeld maar een fingerprint bevat onder meer informatie over de tijd wanneer het voorkomt in het audiofragment. Wanneer verschillende fingerprints overeenkomen kunnen we deze informatie gebruiken om de latency tussen de opnames te bepalen.

Eigenlijk is deze methode nog niet echt nauwkeurig genoeg. Een resultaat tot op één milliseconde nauwkeurig is zeer gewenst. Daarom wordt het resultaat dat bepaalt is met het accoustic fingerprinting nog verfijnt met een tweede methode namelijk door het berekenen van de kruiscovariantie. Dit is een methode waarbij de gelijkenis tussen twee signalen berekent wordt. Aan de mate van gelijkenis wordt een bepaalde score toegekend. Door twee opnames over elkaar te verschuiven en telkens de kruiscovariantie te berekenen kunnen we zeer precies de latency tussen beide audiofragmenten bepalen. Deze berekening is zeer traag, daarom gebruiken we eerst accoustic fingerprinting om tot een ruw resultaat te komen. Vervolgens kan de latency verfijnt worden door de kruiscovariantie te berekenen.

Op dit moment moeten deze methode's na een experiment handmatig op de geluidsopnames worden uitgevoerd. Dit is niet alleen erg ongebruiksvriendelijk maar ook zeer tijdrovend. Daarom heb ik de opdracht gekregen om met behulp van deze methodes een real-time systeem te bouwen. Dit systeem moet in staat zijn om tijdens het experiment de latency onmiddelijk te bepalen en de synchronisatie direct uit te voeren. Hiervoor heb ik verschillende wijzingen moeten aanbrengen aan de synchronisatiealgoritmes. Ook is het de bedoeling dat ik een plugin schrijf voor een audio verwerking software pakket dat het synchroniseren volledig automatiseert voor de onderzoekers van het IPEM.