



Virtual LSD, een immersieve audiovisuele ervaring

door

Robby WAUTERS

Promotor : Prof. dr. ir. E. DUVAL

Proefschrift ingediend tot
het behalen van de graad van
Master in de toegepaste informatica

Academiejaar 2011–2012

© Copyright by K.U.Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wendt u tot het Departement Computerwetenschappen, Celestijnenlaan 200A, 3001 Leuven, (016) 32 77 00 of via email: info@cs.kuleuven.be.

Voorafgaande schriftelijke toestemming van de promotor(en) is eveneens vereist voor het aanwenden van de in dit afstudeerwerk beschreven (originele) methoden, producten en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Faculteit Wetenschappen
Departement Computerwetenschappen
Celestijnenlaan 200A
3001 Leuven
(016) 32 77 00

K.U. Leuven
Academiejaar 2011–2012

Voornaam en naam student : Robby WAUTERS

Titel :

Virtual LSD, een immersieve audiovisuele ervaring

Engelse vertaling :

Virtual LSD, audiovisual immersion

Korte inhoud :

Deze masterproef beschrijft een Augmented Reality applicatie, Virtual LSD, die de opkomst van AR technologie gebruikt om muziekvisualisatiesoftware naar een doelgroep van mobiele computergebruikers te brengen. De applicatie visualiseert de muziek van de gebruiker door zijn omgeving te integreren in de visualisatie. Om dit te verwezenlijken wordt eerst een architectuur opgesteld die de basis vormt voor de Virtual LSD applicatie. Deze architectuur brengt drie onderzoeksdomeinen samen; Music Information Retrieval, Computer Visie en Visualisatie. Door de combinatie van deze domeinen is het mogelijk om een audiovisuele AR applicatie te creëren. Deze applicatie steunt op een set van apparaten die het toelaten om de gebruiker onder te dompelen in een audiovisuele ervaring. Voor de Virtual LSD applicatie wordt gebruik gemaakt van een AR bril, een laptop en een webcam. Met deze architectuur en hardware-opstelling ontwikkelen we twee prototypes die aan de hand van onderzoek naar immersie en muziekbeleving het nut van de applicatie aantonen. Om de graad van immersie te meten hebben we nood aan een nuttige evaluatiemethode. In de evaluaties van de prototypes worden meerdere methodes op de proef gesteld. Hieruit zal blijken dat een eigen ontwikkelde questionnaire de meest pertinente informatie biedt. Uit de evaluatiesessies kunnen we besluiten dat de applicatie een effectieve meerwaarde biedt aan het beluisteren van muziek en dat een zekere vorm van immersie aanwezig is. Deze bevindingen tonen de toegevoegde waarde die een dergelijke applicatie kan bieden voor een steeds groeiende doelgroep van mobiele computergebruikers. De groeiende publieke en commerciële interesse in AR, toont aan dat er toekomstperspectieven zijn voor applicaties zoals Virtual LSD.

Proefschrift ingediend tot het behalen van de graad van
Master in de toegepaste informatica

Promotor : Prof. dr. ir. E. Duval

Assessoren : Prof. dr. T. Holvoet
F. Van Assche

Begeleider : Dr. J. Klerkx

Voorwoord

Deze masterproef is het sluitstuk van mijn vijfjarige loopbaan in hoger en universitair onderwijs. Deze studies werden verrijkt met de steun en raad van meerdere personen.

Hierbij wil ik de tijd nemen om hen te bedanken.

Allereerst wil ik mijn promotor, professor Erik Duval, bedanken voor zijn geloof in deze masterproef en zijn waardevolle opmerkingen tijdens zowel de tussentijdse presentaties als de evaluatiesessies. Ik wens ook Joris Klerkx, mijn begeleider, in grote mate te bedanken voor zijn kritische feedback die hij gedurende het hele jaar gaf. Zijn bijdrage was essentieel voor het welslagen van deze masterproef.

Ik heb in deze masterproef het privilege gehad om met nieuwe en spannende technologieën te werken. Mijn dank gaat dan ook uit naar de HCI groep voor het uitlenen van de AR bril en de onderzoeksgroep Neurofysiologie voor het uitlenen van de EEG headset. Daarnaast wil ik ook Arne Robben bedanken voor de ondersteuning die hij geboden heeft bij het meten van hersengolffrequenties.

Ook de personen die deelnamen aan de evaluatiesessies wil ik hierbij bedanken. Hun oprechte verklaringen en opmerkingen waren van onschatbare waarde bij het ontwikkelen en beoordelen van de Virtual LSD applicatie.

Tenslotte wil ik mijn dankbaarheid overbrengen aan mijn familie, partner en vrienden voor hun onvoorwaardelijke steun. In het bijzonder wil ik mijn broer en schoonzus bedanken voor het verbeteren en controleren van de masterproeftekst. Ook ben ik Koen Boncquet dankbaar voor de talloze discussies en brainstormsessies gedurende dit jaar. Zonder hun steun zou deze masterproef een onmogelijke taak geweest zijn.

Robby Wauters

Abstract

Deze masterproef beschrijft een Augmented Reality applicatie, Virtual LSD, die de opkomst van AR technologie gebruikt om muziekvisualisatiesoftware naar een doelgroep van mobiele computergebruikers te brengen. De applicatie visualiseert de muziek van de gebruiker door zijn omgeving te integreren in de visualisatie. Om dit te verwezenlijken wordt eerst een architectuur opgesteld die de basis vormt voor de Virtual LSD applicatie. Deze architectuur brengt drie onderzoeksdomeinen samen; Music Information Retrieval, Computer Visie en Visualisatie. Door de combinatie van deze domeinen is het mogelijk om een audiovisuele AR applicatie te creëren. Deze applicatie steunt op een set van apparaten die het toelaten om de gebruiker onder te dompelen in een audiovisuele ervaring. Voor de Virtual LSD applicatie wordt gebruik gemaakt van een AR bril, een laptop en een webcam. Met deze architectuur en hardware-opstelling ontwikkelen we twee prototypes die aan de hand van onderzoek naar immersie en muziekbeleving het nut van de applicatie aantonen. Om de graad van immersie te meten hebben we nood aan een nuttige evaluatiemethode. In de evaluaties van de prototypes worden meerdere methodes op de proef gesteld. Hieruit zal blijken dat een eigen ontwikkelde questionnaire de meest pertinente informatie biedt. Uit de evaluatiesessies kunnen we besluiten dat de applicatie een effectieve meerwaarde biedt aan het beluisteren van muziek en dat een zekere vorm van immersie aanwezig is. Deze bevindingen tonen de toegevoegde waarde die een dergelijke applicatie kan bieden voor een steeds groeiende doelgroep van mobiele computergebruikers. De groeiende publieke en commerciële interesse in AR, toont aan dat er toekomstperspectieven zijn voor applicaties zoals Virtual LSD.

Voor beeldmateriaal en code: <http://rwd.snakeflash.com/blog/>

Inhoudsopgave

Voorwoord	i
Abstract	ii
Inhoudsopgave	iii
Lijst van figuren en tabellen	vi
1 Inleiding	1
1.1 Situering van het onderwerp	1
1.2 Titelverklaring	2
1.3 Idee	3
1.4 Doelstellingen	5
1.5 Structuur van de tekst	6
2 Ontwerp	7
2.1 Ontwikkelmethodologie	7
2.2 Architectuur van de applicatie	7
2.3 Hardware	8
2.3.1 Opstelling	8
2.3.2 Draagbare display - AR bril	9
2.3.3 Camera	9
2.3.4 Laptop	10
2.3.5 EEG headset	11
2.4 Gerelateerde toepassingen	11
2.4.1 Muse'ic visualizer	11
2.4.2 BEP360	12
2.4.3 ClayVision	13
2.5 Literatuurstudie rond immersie	13
2.5.1 Duiding van immersie	14
2.5.2 Flow	14
2.5.3 Cognitive Absorption	15
2.5.4 Presence	16

2.5.5	Eigen definitie van immersie	16
3	Music Information Retrieval	18
3.1	Literatuurstudie	18
3.1.1	Domein	18
3.1.2	Basisoperatoren	20
3.1.3	Audio features	21
3.1.4	Muziek, emotie en kleur	22
3.2	Raamwerken	23
3.2.1	Marsyas	23
3.2.2	ALF	23
3.2.3	YAAFE	24
3.2.4	MIRtoolbox	24
3.2.5	jMIR	24
3.2.6	Vergelijking	25
3.3	Implementatie	28
3.3.1	Audio features extraheren	28
3.3.2	Muziek afspelen	30
3.3.3	Audio features aanspreken	31
4	Computer visie	34
4.1	Literatuurstudie	34
4.1.1	Detecteren van <i>interest points</i>	34
4.1.2	Detecteren van contouren	35
4.1.3	Toepassingen	37
4.2	Raamwerken	38
4.2.1	OpenCV	39
4.2.2	BoofCV	40
4.3	Implementatie	41
4.3.1	Webcamfeed ophalen	41
4.3.2	<i>Interest points</i> detecteren	44
4.3.3	Contourdetectie	44
4.3.4	Bewegingsdetectie	45
4.3.5	Huiddetectie	45
5	Visualisatie	46
5.1	Situering	46
5.2	Raamwerken	47
5.2.1	Adobe Flash	47
5.2.2	Processing	47
5.2.3	OpenFrameworks	48
5.2.4	Vergelijking	48
5.3	Implementatie	49

5.3.1	Tekenen met Processing	49
5.3.2	Basisstructuur van een visualisatie	50
5.3.3	Geïmplementeerde visualisaties	51
6	Evaluatie	57
6.1	Eigen definitie van immersie	57
6.2	Methodes voor het evalueren van beleving en immersie	58
6.3	Het ontwikkelen van de questionnaire	59
6.4	Eerste evaluatiesessie	60
6.4.1	Evaluatieverloop	60
6.4.2	Resultaten	61
6.4.3	Conclusie	64
6.5	Tweede evaluatiesessie	65
6.5.1	Evaluatieverloop	65
6.5.2	Resultaten	66
6.5.3	Conclusie	69
7	Besluit	70
7.1	Doelstellingen	70
7.2	Kritische reflectie	73
7.3	Mogelijke toekomstige uitbreidingen	74
A	Architectuur van de Virtual LSD applicatie	75
B	Questionnaire	77
C	Boxplotgrafiek van resultaten uit eerste evaluatiesessie	79
D	Boxplotgrafiek van resultaten uit tweede evaluatiesessie	81
E	Poster	83

Lijst van figuren en tabellen

Figuren

2.1	De architectuur van de Virtual LSD applicatie, die de drie grote onderzoeksdomeinen schetst; <i>Music Information Retrieval, Computer Visie en Visualisatie</i>	8
2.2	De Vuzix Wrap 920 videobrill met bijbehorende VGA-adapter. <i>Bron: Vuzix</i>	9
2.3	Logitech C200 webcam, vastgemaakt aan de videobrill.	10
2.4	De nieuwe flexibele webcam, vastgemaakt aan de videobrill	10
2.5	De EEG headset in de volledige opstelling.	11
2.6	De Muse'ic visualizer iPhone app. <i>Bron: Keigo Oyamada, Salyu</i>	12
2.7	De BEP360 iPhone app. <i>Bron: Will.i.apps</i>	13
3.1	Grafische gebruikersinterface van jAudio	29
3.2	ACE XML exportbestand waarin de audio features beschreven worden . .	31
3.3	Algoritme om real-time audio features op te vragen	32
3.4	Audio features geplot in grafieken. De waarde van een audio feature wordt getekend langs de y-as. De x-as geeft het tijdsverloop aan.	33
4.1	Regio's die gevonden kunnen worden door Harris Corner detectie-algoritme. Er kunnen vlakke regio's (links), randen (midden) en hoekpunten (rechts) herkend worden. <i>Bron: Czech Technical Univeristy Prague</i>	35
4.2	Het Harris Corner detectie-algoritme toegepast op een beeld (links) en de gedetecteerde hoeken (rechts).	36
4.3	De meerdere fases van het <i>Canny</i> detectie-algoritme. <i>Bron: Moeslund[38]</i> .	37
4.4	Contouren die gedetecteerd en gevisualiseerd worden met het OpenCV-raamwerk	39
4.5	Het detecteren van <i>interest points</i> met het BoofCV-raamwerk.	40
4.6	Vergelijkende studie tussen BoofCV en OpenCV. OpenCV scoort beter op <i>low level</i> algoritmes, terwijl Boofcv beter presteert <i>high level</i> operaties zoals het detecteren van <i>interest points</i> . De snelheden zijn relatief uitgedrukt, waardoor een hogere score beter is. <i>Bron: www.boofcv.org</i>	41
4.7	Algoritme dat webcam-beelden tekent op het scherm met behulp van Processing en JMF.	43

4.8	Algoritme dat webcam-beelden tekent op het scherm met behulp van Processing en GSVideo.	44
4.9	Gedetecteerde <i>interest points</i> worden gevisualiseerd in verscheidene kleuren.	44
4.10	Detectie van contouren in een beeld met behulp van het <i>Canny</i> detectie-algoritme uit het BoofCV-raamwerk.	45
5.1	Basisapplicatie met gebruik van Processing raamwerk	50
5.2	Overzicht van het Fish-eye effect. Het originele beeld (a), het beeld waar een Whirl-effect(b) en een Pinch-effect(c) is op uitgevoerd	53
5.3	De visualisaties uit het eerste prototype van de Virtual LSD applicatie. . .	55
5.4	De visualisaties uit het tweede prototype van de Virtual LSD applicatie. . .	56
6.1	Samengevoegde Boxplotgrafiek	62
6.2	Boxplotgrafiek per kerncomponent	63
6.3	Boxplotgrafiek per kernbegrip	66
6.4	Antwoorden op de interviewvraag <i>Klonk de muziek intenser tijdens het gebruik van de Virtual LSD applicatie?</i>	67
6.5	Calibratiemetingen van testgebruikers met Mind WorkStation software en de Emotiv Epoc headset.	68
6.6	Metingen tijdens de Virtual LSD applicatie met Mind WorkStation software en de Emotiv Epoc headset.	68

Tabellen

3.1	Overzicht van audio features per raamwerk	25
3.2	Overzicht van muziekanalyse-raamwerken	26

Hoofdstuk 1

Inleiding

1.1 Situering van het onderwerp

Sinds de jaren '90 zijn visualisaties van muziek populair geworden door de komst van *media player* software. Softwarepakketten zoals Winamp, Windows Media Player en iTunes werden massaal gebruikt door de stijgende bereikbaarheid van digitale muziek via het wereldwijde web. Deze software kon naast muziek afspelen en organiseren, ook visualisaties genereren bij de muziek. De modules die de visualisaties voor hun rekening namen (e.g. Milkdrop of G-force), genereerden abstracte beelden die leken te reageren op de muziek.

De populariteit van deze visualisatiesoftware is in de laatste tien jaar sterk gedaald. Door de komst van draagbare computers, zoals MP3-spelers, tablets en smartphones, zijn computergebruikers zeer mobiel geworden. Sinds de lancering van de populaire iPod MP3-speler werd steeds meer muziek onderweg beluisterd. Tegenwoordig zijn er aanzienlijk veel mensen die naar muziek luisteren via een smartphone of MP3-speler terwijl ze op de bus zitten, fietsen, wandelen of joggen. De bestaande muziekvisualisatiesoftware bereikt echter deze doelgroep niet. Een mogelijkheid is om muziekvisualisaties te integreren in de omgeving van deze gebruikers, zodat ze nog steeds kunnen navigeren in de echte wereld. De focus ligt zo op de muziek en de bijpassende visualisatie, terwijl de gebruiker nog steeds mobiel is.

Een technologie die een oplossing kan bieden voor dit probleem is Augmented Reality (AR). AR is het rijker maken van een bestaande omgeving door virtuele elementen te integreren in het beeld van de gebruiker. Sinds de ontwikkeling van het eerste AR prototype door Ivan Sutherland[51] in 1968, is er steeds een sterke interesse geweest voor AR en Virtual Reality (VR) in de onderzoekswereld. Pas in de jaren '90 is echter de term Augmented Reality voor het eerst gevallen gevallen bij een onderzoeksgroep van Boeing Corporation. Deze onderzoeksgroep ontwikkelde een systeem die hun werknemers zou moeten ondersteunen tijdens het assembleren van vliegtuigonderdelen. Tot op heden worden dergelijke systemen nog steeds gebruikt in de industrie. AR wordt ook toegepast in militaire systemen, waarbij men extra informatie omtrent strategie en omgeving toont aan militairen. Navigatiesystemen en toepassingen die dokters ondersteunen bij medische ingrepen maken

ook (vaak) gebruik van AR. Tot voor kort waren er echter weinig concrete entertainment-systemen beschikbaar die AR gebruiken[56]. In de laatste vijf jaar is er een steeds sterker groeiende publieke interesse voor AR. Een factor die hier (nog steeds) een positieve invloed op heeft, is de hype rond 3D films en games. Verscheidene bedrijven (e.g. Vuzix en Sony) hebben in de laatste vijf jaar *low-cost* HMD¹-apparaten op de markt gebracht, die toelaten om 3D films en games thuis te ervaren. In april 2012 heeft Google voor oproer gezorgd met de aankondiging van Project Glass[40]. Hiermee heeft AR heel wat populariteit verkregen bij het grote publiek.

In deze masterproef wordt een AR applicatie beschreven die de opkomst van een 'nieuwe' technologie gebruikt om muziekvisualisatiesoftware naar een mobiele doelgroep te brengen. De applicatie zou de muziek van een mobiele gebruiker moeten kunnen visualiseren door de omgeving te integreren in de visualisatie.

1.2 Titelverklaring

De titel van deze thesis luidt “Virtual LSD, een immersieve audiovisuele ervaring”. *Virtual LSD* is de naam van de muziekvisualisatiesoftware die voor een mobiele doelgroep gemaakt werd in deze masterproef. De term *LSD* verwijst naar de hallucinogene drug² die ontwikkeld werd in 1938 door Albert Hofmann. Bij de ontdekking van de drug werd een enorme versterking in verbeeldingskracht waargenomen, die zorgde voor een veranderde waarneming van de omgeving[19]. In de jaren '60 en '70 werd LSD populair bij festivalgangers, die beweerden dat de muziek hun omgeving veranderde en hen onderdompelde in een nieuwe wereld[15]. Gelijkaardig aan deze hallucinogene drug, zal de applicatie die beschreven wordt in de masterproef de omgeving van een gebruiker aanpassen op basis van muziek. Om aan te geven dat het gaat over een puur virtuele representatie van de drug, werd de term *virtual* geïntroduceerd in de naam.

Een *audiovisuele ervaring* refereert naar de combinatie van de auditieve en visuele zintuigen die geprikkeld worden in de applicatie. Zowel de muziek als de omgeving die hiermee aangepast wordt, zijn de meest essentiële aspecten binnen de Virtual LSD applicatie.

De term *immersie* beschrijft een psychologische toestand waar een persoon zich in een andere wereld waant. Een letterlijke vertaling van deze term is *onderdompeling*. Net zoals bij de hallucinogene drug zal een gebruiker van de Virtual LSD applicatie ondergedompeld worden in een nieuwe wereld. De term wordt verder uitgebreid besproken in sectie 2.5. De introductie van immersie in de titel, benadrukt de aandacht dat hieraan gespendeerd werd in deze masterproef. De Virtual LSD applicatie werd gemaakt om de immersie en muziekbeleving van de gebruiker te kunnen bestuderen.

¹Head-Mounted Display: Een display dat gedragen kan worden op het hoofd.

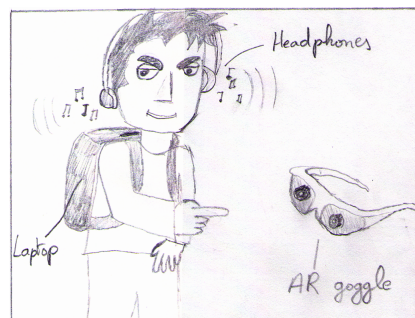
²LSD: LySergeenzuurDi-ethylamide

1.3 Idee

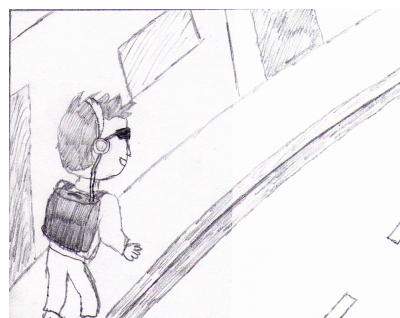
Deze sectie beschrijft het idee van de Virtual LSD applicatie aan de hand van een storyboard. Een storyboard is een opeenvolging van visuele schetsen, aangevuld met verduidelijkende informatie[20]. Bovendien maakt de ontwikkeling van een storyboard het mogelijk om goedkoop en snel een visuele en overzichtelijke voorstelling van het idee te presenteren.

Op stap met Steve

Steve is een muziek en technologie fanaat. Hij houdt van feesten en besteedt uren aan het luisteren naar en het maken van muziek thuis. Onlangs heeft Steve een manier gevonden om volledig ondergedompeld worden in een muzikale ervaring. Hij heeft een combinatie van apparaten verbonden met zijn laptop, die een AR visualisatie programma uitvoert. Gewapend met een laptop op zijn rug, koptelefoon stevig op zijn hoofd, heeft hij enkel nog een AR-bril nodig. Deze bril toont een video-feed die zijn normaal zicht vertegenwoordigt. Deze video-feed wordt overgebracht naar lcd-displays in de bril. Zo kan Steve zijn omgeving zien alsof hij een normale bril zou dragen.

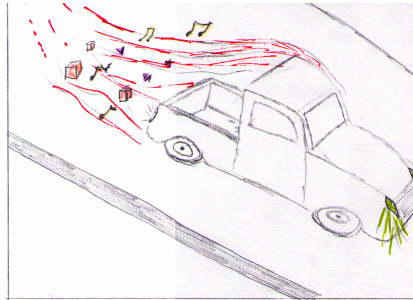


Steve verbindt zijn hoofdtelefoon en een AR-bril aan zijn laptop en start de Virtual LSD-applicatie. Het programma is gekoppeld aan een muziekspeler waarmee Steve elk nummer kan kiezen en afspelen dat hij wil. Na de opstelling, maakt Steve een wandeling in de stad.

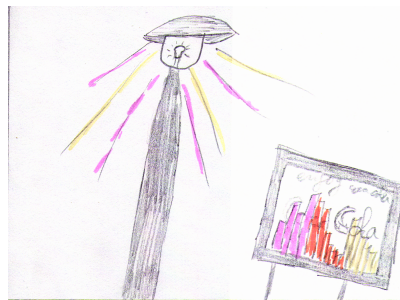


De Virtual LSD-applicatie zorgt ervoor dat Steve volledig wordt ondergedompeld in een audiovisuele ervaring. De verlichting van de straat verandert volgens het ritme van de muziek. De wereld om hem heen krijgt een hele nieuwe vorm van belichting en kleur.

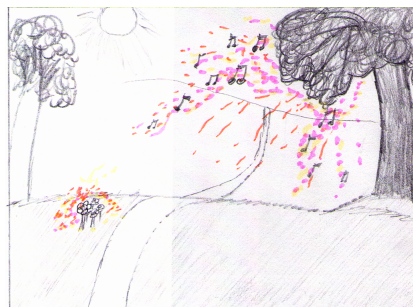
Wanneer Steve een voorbijrijdende auto ziet, worden een enorme hoeveelheid deeltjes gegenereerd in het spoor van het bewegende voertuig. Heldere, opvallende, kleurrijke muzieknoten, kubussen en andere deeltjes reageren op de muziek.



Het straatlicht dat Steve voorbij wandelt lijkt verschillende kleuren uit te stralen op basis van de muziek die wordt afgespeeld. Wanneer hij een reclamebord opmerkt, ziet hij een visualisatie van de audio-amplitudes. Hij zit ondertussen helemaal in de sfeer van zijn favoriete nummers die door zijn koptelefoon te horen zijn.



Aan de rand van de stad, betreedt Steve het bos. Hij gelooft zijn ogen niet wanneer hij het bos tot leven ziet komen. Deeltjeswolken die ontstaan uit bomen en bloemen, bewegen op de muziek die hij hoort.



Als Steve thuis komt, kan hij nog steeds niet geloven hoe intens hij de muziek heeft ervaren. Hij voelde de sfeer en het genre van de nummers op een geheel nieuwe manier. Elk nummer lijkt een uniek effect te hebben op zijn omgeving. Steve sms't al zijn vrienden om deze geweldige ervaring zelf ook eens proberen...

1.4 Doelstellingen

In deze masterproef wordt een *proof-of-concept* AR applicatie gemaakt die muziek visualiseert in de omgeving van de gebruiker. Met deze applicatie wordt muziekbeleving en immersie onderzocht. De verscheidene stappen en deeloplossingen die nodig zijn om dit te bereiken worden hier overlopen als doelstellingen van deze masterproef.

Onderzoeken van de architectuur Een eerste doel in deze thesis is het onderzoeken waaruit een audiovisuele AR applicatie bestaat. Er moet een globale architectuur opgesteld worden, waarmee een dergelijke applicatie gebouwd kan worden. Binnen deze architectuur zal een keuze moeten gemaakt worden tussen raamwerken die combineerbaar zijn en de juiste functionaliteit aanbieden.

Onderzoeken van een hardware-opstelling Om een systeem te ontwikkelen dat gebruik maakt van audio, visualisaties en AR is een set van bepaalde apparaten nodig. Binnen deze masterproef zal een opstelling moeten onderzocht worden die in staat is om een audiovisueel AR systeem functioneel te maken.

Ontwikkelen van een *Proof-of-concept* AR applicatie Aan de hand van de vooropgestelde architectuur willen we op een iteratieve manier prototypes ontwikkelen die in staat zijn om muziek te visualiseren in de omgeving van de gebruiker.

Integratie van de omgeving onderzoeken In een volgende stap zou de integratie van de omgeving in de applicatie moeten onderzocht worden, door de immersie te evalueren van de gebruiker. Immersie geeft de mate aan dat een gebruiker zich in een virtuele wereld waant.

Onderzoeken van methodes om immersie te meten Om immersie bij een gebruiker te kunnen meten, moet een methode gevonden worden die hier een oordeel over kan vellen.

Aantonen van het nut We willen dat de ontwikkelde applicatie nuttig is en een toegevoegde waarde kan betekenen in bepaalde scenario's.

Aantonen van de praktische haalbaarheid Tenslotte willen we ook de haalbaarheid ingeschat worden om een dergelijke applicatie commercieel te maken.

1.5 Structuur van de tekst

De structuur van deze masterproef werd opgebouwd rond de Virtual LSD applicatie, waarmee de immersie en beleving van de gebruiker werden bestudeerd. In hoofdstuk 2 wordt het ontwerp besproken van de Virtual LSD applicatie. Hierin wordt een architectuur geschetst van de applicatie en de hardware die er voor nodig is. Hierna wordt aan immersie een definitie gegeven die gebruikt werd om de Virtual LSD applicatie te evalueren.

De hoofdstukken 3, 4 en 5 bespreken de verschillende onderzoeksdomeinen die aan bod komen in de Virtual LSD applicatie. In deze hoofdstukken wordt steeds de literatuur bestudeerd die een basis vormt om in een later stadium de praktische en technische aspecten van het onderzoeksdomein te kunnen bespreken. Voor de bespreking van de implementatie, worden raamwerken vergeleken en uitgekozen die nuttig leken binnen het domein.

De Virtual LSD applicatie steunt op drie sterk verschillende onderzoeksdomeinen. Het eerste is *Music Information Retrieval*, dat het ophalen van informatie uit muziek beschrijft in hoofdstuk 3. In hoofdstuk 4 komt Computer Visie aan de beurt, dat beschrijft hoe we informatie halen uit de omgeving van de gebruiker. Tenslotte wordt in hoofdstuk 5 de visualisaties besproken die gebruik maken van de vorige twee onderzoeksdomeinen.

Na de uitgebreide bespreking van elk onderzoeksdomein wordt in hoofdstuk 6 iets meer gezegd over de evaluatie van de prototypes. Eerst wordt er een gepaste evaluatiemethode gezocht waarmee we immersie bij de gebruiker kunnen evalueren. In een volgende stap worden twee evaluatiesessies gehouden die hier uitvoerig besproken worden.

Deze masterproef wordt afgesloten met een besluit waarin de vooropgestelde doelen getoetst worden aan de behaalde resultaten. In dit laatste hoofdstuk zal ook een kritische reflectie gegeven worden over de aanpak van deze masterproef. Tenslotte worden enkele pistes aangehaald die kunnen leiden tot verder onderzoek.

Hoofdstuk 2

Ontwerp

In dit hoofdstuk wordt het ontwerp van de Virtual LSD applicatie meer uitgelegd. Hierbij wordt de ontwikkelmethodologie besproken die gehanteerd werd. Daarna wordt de globale architectuur van de applicatie opgesteld. Hiermee zal de applicatie opgebouwd worden. Een audiovisualisatie applicatie die gebruik maakt van AR heeft bepaalde apparaten nodig, waaraan we in dit hoofdstuk aandacht zullen besteden. Naast deze zaken worden ook gerelateerde toepassingen aangehaald. We besluiten met een literatuurstudie rond immersie. Dit concept wordt vaak doorheen de masterproef gebruikt, waardoor het noodzakelijk is dat we het concept in dit hoofdstuk duidelijk definiëren.

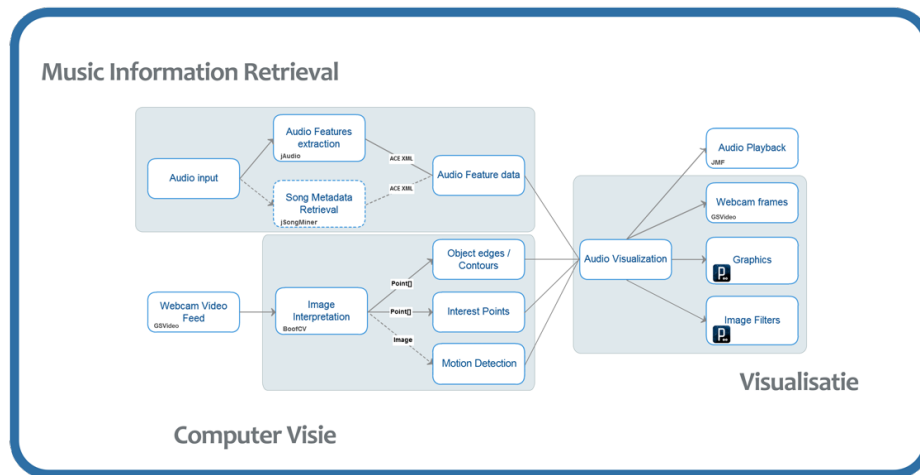
2.1 Ontwikkelmethodologie

Om de Virtual LSD applicatie te ontwikkelen werd gebruik gemaakt van *rapid prototyping*. We zullen op een iteratieve manier prototypes ontwikkelen. Na de ontwikkeling, zal het prototype getest worden door middel van evaluaties met gebruikers. Met de bevindingen van elke evaluatie, zullen we het prototype verbeteren of uitbreiden. In deze masterproef werden twee opeenvolgende prototypes gemaakt die elk geëvalueerd werden.

2.2 Architectuur van de applicatie

De Virtual LSD applicatie bestaat uit meerdere onderzoeksdomeinen. De theoretische en technische aspecten worden per onderzoeksdomein uitgelegd in hoofdstuk 3, 4 en 5. Alvorens een specifiek onderzoeksdomein verder uit te diepen, wordt hier de globale architectuur van de applicatie geschetst. Figuur 2.1 toont de verschillende componenten van de Virtual LSD applicatie. Zonder te diep in te gaan op de onderliggende raamwerken, kunnen we deze componenten splitsen in drie grote delen. Het eerste domein is *Music Information Retrieval*. Dit domein bevat de componenten die instaan voor het halen van informatie uit muziek. Een ander domein, *Computer Visie*, bevat dan weer degene die instaan voor het halen van informatie uit de omgeving van de gebruiker. Als laatste is er *Visualisatie*, dat

de informatie van de vorige domeinen zal gebruiken om zinvolle beelden te genereren op het scherm. In bijlage A wordt deze architectuur in volledige grootte opgenomen.



Figuur 2.1: De architectuur van de Virtual LSD applicatie, die de drie grote onderzoeksdomeinen schetst; *Music Information Retrieval*, *Computer Visie* en *Visualisatie*

2.3 Hardware

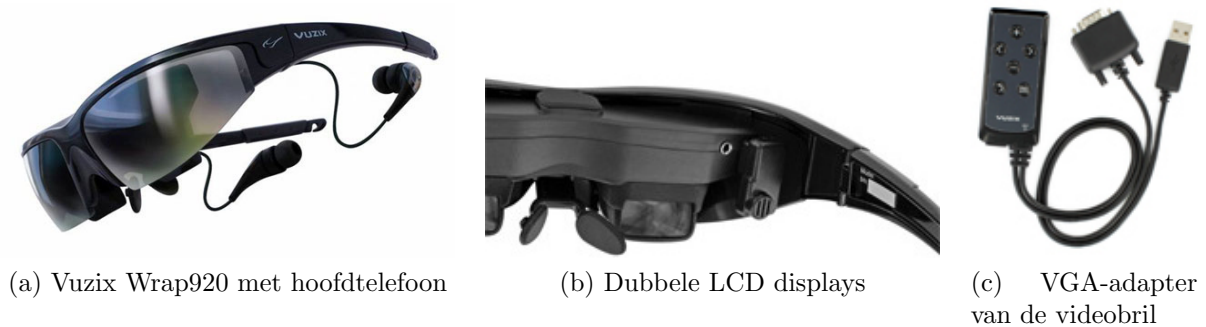
2.3.1 Opstelling

Zoals vermeld in het storyboard (1.3), heeft de gebruiker van de Virtual LSD applicatie enkele apparaten nodig. Een gebruiker heeft een laptop (uitbreidbaar naar smartphone), een hoofdtelefoon en een AR bril nodig. De AR bril wordt hier gezien als een videobrill die bovendien in staat is om de omgeving te registreren via ingebouwde camera's. Een voorbeeld van dergelijke bril is de Vuzix Wrap 920AR¹. De hoge kostprijs van deze AR bril, leek een struikelblok te zijn om de bril te gebruiken in deze masterproef. Er was echter een gelijkaardige videobrill aanwezig op het departement (CW KULeuven) met dezelfde specificaties, maar zonder de camera's. Door zelf een camera toe te voegen aan de opstelling, verkrijgen we een gelijkaardig apparaat met een substantieel lagere kost. Bovendien is een hoofdtelefoon geïntegreerd in de bril. De uiteindelijke opstelling van de Virtual LSD applicatie bestaat vervolgens uit:

- Draagbare display of AR bril
- Camera
- Laptop

De komende secties beschrijven de gebruikte apparaten en mogelijke alternatieven.

¹Vuzix Wrap 920AR, http://vuzix.com/consumer/products_wrap920ar.html



Figuur 2.2: De Vuzix Wrap 920 videobril met bijbehorende VGA-adapter. *Bron: Vuzix*

2.3.2 Draagbare display - AR bril

De draagbare display die gebruikt werd in de Virtual LSD applicatie is de Vuzix Wrap 920. Zoals eerder vermeld heeft deze videobril geen ingebouwde camera's, maar komt verder sterk overeen met de Vuzix Wrap 920AR. De bril met een ingebouwde hoofdtelefoon is te zien in figuur 2.2a. De makers hebben getracht om de bril zo onopmerkelijk mogelijk te maken, door het te maskeren als een zonnebril. Door de hoge plaats op de neus, is deze poging echter niet geslaagd. Binnenin de bril zitten twee LCD displays (figuur 2.2b), die elk een resolutie hebben 640 op 480 pixels. Doordat deze displays zo dicht tegen de ogen staan, heeft de gebruiker de indruk dat hij naar een 67" scherm kijkt vanop 3 meter afstand. Via een VGA-adapter (figuur 2.2c) is het mogelijk om de videobril te koppelen aan een laptop. De maximale resolutie die door de VGA-adapter kan worden doorgestuurd is 1024 op 768 pixels. Uit de evaluaties bleek dat deze resolutie geen beperking was voor de Virtual LSD applicatie. Om een tekstdocument te lezen met deze bril, blijkt de resolutie wel onvoldoende te zijn. Aan de VGA-adapter is ook een USB-uitgang aanwezig die instaat voor de stroomtoevoer en het geluid. Door de USB-kabel te connecteren met de laptop, wordt het geluid van het besturingssysteem doorgestuurd naar de ingebouwde hoofdtelefoon van de bril.

2.3.3 Camera

Om een alternatief te bieden voor de ingebouwde camera's, werd een webcam gekoppeld aan de bril. In de eerste testen van de Virtual LSD applicatie werd een Logitech C200 webcam gebruikt. In figuur 2.3 is te zien dat de bolvormige webcam bovenop de bril rust. De vorm van de camera, zorgde er voor dat de bril oncomfortabel zat en de hele opstelling onstabiel was. Bovendien was de kwaliteit van de webcam te laag om behoorlijke beelden te produceren.

Om de problemen van de vorige camera op te lossen, werd een nieuwe webcam aangekocht. De keuze viel op een kleine webcam met een flexibele 'nek'. In figuur 2.4 zijn foto's te zien van deze webcam, vastgemaakt aan de videobril. Deze webcam produceert een scherper



Figuur 2.3: Logitech C200 webcam, vastgemaakt aan de videobrill.

beeld, heeft dezelfde kleur en stijl van de bril en zorgt voor geen bijkomend ongemak bij het dragen van de bril. De webcam werd vastgemaakt langs de linkerzijde van de bril op een relatief primitieve manier (figuur 2.4c). Tijdens de evaluaties ondervonden de gebruikers geen ongemak van de webcam.



(a) Bovenaanzicht van videobrill en webcam

(b) vooraanzicht van videobrill en webcam

(c) Aanhechten van webcam aan videobrill

Figuur 2.4: De nieuwe flexibele webcam, vastgemaakt aan de videobrill

2.3.4 Laptop

De laptop die gebruikt werd om de Virtual LSD applicatie te draaien, was een MSI GX700. Deze vijf jaar oude laptop heeft een Intel Core2Duo T7300 processor met 2Gb aan werkgeheugen. Uit de evaluaties is gebleken dat de laptop vaak tekortschoot om de visualisaties vlot te tonen. Door optimalisaties aan de software door te voeren, is de performantie van het tweede prototype sterk verbeterd.

In de opstelling zou de laptop kunnen vervangen worden door een smartphone. Het gebruik van een smartphone zou het realisme van de applicatie vergroten. Het zou echter ook de implementatie en *debuggen* moeilijk maken. Bovendien zijn de specificaties van *state of the art* smartphones vergelijkbaar met de specificaties van de gebruikte laptop. Om het implementeren, testen en uitvoeren van de Virtual LSD applicatie eenvoudiger te maken, werd een laptop verkozen boven een smartphone.

2.3.5 EEG headset

In de tweede evaluatiesessie werd getracht om emotionele reacties te registreren met hersengolffrequenties. Om dit te verwezenlijken werd een EEG² headset gebruikt die tijdens de applicatie gedragen werd. De gekozen EEG headset was de Emotiv Epoc. De Emotiv Epoc is een draadloze headset die hersengolffrequenties registreert en verwerkt (figuur 2.5a). Het maakt gebruik van een set van (14) sensoren die elektrische signalen opvangen die door de hersenen worden geproduceerd. Deze sensoren worden op verscheidene plaatsen op de schedel geplaatst, zodat een breed bereik van hersengolven kan gemeten worden. Met deze EEG headset kunnen gedachten, gevoelens en gezichtsexpressies van de gebruiker worden geregistreerd³. De volledige opstelling inclusief EEG headset is te zien in figuur 2.5b. Het gebruik van de videobrill samen met de EEG headset is mogelijk, maar zeker en vast niet comfortabel. Tijdens de tweede evaluatiesessie klaagden verschillende testpersonen dat het gebruik van beide voor ongemak zorgde.



(a) De Emotiv Epoc headset. Bron: Emotiv



(b) De volledige opstelling inclusief Emotiv Epoc

Figuur 2.5: De EEG headset in de volledige opstelling.

2.4 Gerelateerde toepassingen

2.4.1 Muse'ic visualizer

Muse'ic visualizer⁴ is een iPhone app die zeer sterk lijkt op de Virtual LSD applicatie. Het is een muziekvisualisatie app, die gecreëerd werd om het muzieknummer *muse'ic* van *Salyu x Salyu* te promoten. De app verandert de omgeving van de gebruiker in een muziekvideo door allerlei filters toe te passen op het webcam-beeld van de iPhone (2.6c). De iPhone moet voor de gebruiker gehouden worden, terwijl de webcam op de achterkant van de

²EEG: elektro-encefalogram

³Emotiv, <http://www.emotiv.com>

⁴Muse'ic visualizer, http://www.merceddeath.com/sxsw/museic_visualiser/

iPhone de omgeving filmt. Ondanks het verschil in gebruikte hardware, heeft deze app bijgedragen tot het ontwikkelen van de Virtual LSD applicatie. Ze gaf bijvoorbeeld het potentieel aan van een dergelijk idee. De enthousiaste reacties op deze app, gaven blijk van een degelijk nut. Ondanks de sterke gelijkenissen heeft deze app enkele beperkingen die vermeden werden in de Virtual LSD applicatie. Er is bijvoorbeeld geen computer visie aanwezig die extra informatie uit het beeld haalt. Daarenboven wordt er enkel gebruik gemaakt van filters die willekeurig op het scherm getoond worden. De grootste beperking is dat deze app gebouwd is rond een specifiek muziknummer. Er worden bijvoorbeeld beelden getoond van de zangeres (figuur 2.6b).

De Virtual LSD applicatie biedt meer diepgang en mogelijkheden door het gebruik van computer visie en abstracte visualisaties die niet gebonden zijn aan een specifiek muziknummer. Hier worden dus geen beelden (e.g. van een zangeres) getoond die gerelateerd zijn aan één muziknummer.



(a) Hoofdscherm van de Muse'ic app.

(b) Visualisatie in de Muse'ic app.

(c) Gebruik van iPhone camera in de Muse'ic app.

Figuur 2.6: De Muse'ic visualizer iPhone app. *Bron: Keigo Oyamada, Salyu*

2.4.2 BEP360

BEP360 is een iPhone app dat op een alternatieve manier muziekvideo's toont⁵. Er zijn verscheidene mogelijkheden, waaronder het bekijken van een muziekvideo waarin je 360 kan rondraaien (figuur 2.7a). Door de iPhone rond te draaien, beweegt de muziekvideo (die te zien is op het scherm) mee in de overeenkomstige richting. De gebruiker van de iPhone app bevindt zich als het ware in het midden van de muziekvideo. De muziekvideo die getoond wordt, is opgenomen met een set van 16 camera's die samen 360 kunnen filmen (2.7b). Als de gebruiker (met de iPhone voor zich uit) roteert om zijn as, worden steeds beelden getoond van overeenkomstige camera's. Zo wordt een illusie gecreëerd waarbij de gebruiker zich in de muziekvideo waant.

⁵BEP360, <http://www.williapps.com/>

Deze iPhone app toont een alternatieve manier om muziekvideo's te tonen door gebruik te maken van de omgeving van de gebruiker. De Virtual LSD applicatie valt ook binnen deze categorie. Ze kan gebruikt worden om muziekvideo's te tonen in de omgeving van de gebruiker.



(a) De 360 muziekvideo functie van de BEP360 app.



(b) De camera-opstelling dat gebruikt werd om de muziekvideo in 360 te filmen.

Figuur 2.7: De BEP360 iPhone app. *Bron: Will.i.apps*

2.4.3 ClayVision

ClayVision[52] is een navigatiesysteem dat Augmented Reality gebruikt om real-time 3D transformaties uit te voeren op stedelijke objecten. Bestaande gebouwen worden vergroot of vervormd als men deze bekijkt via het navigatiesysteem. Om dit te verwezenlijken hebben de ontwikkelaars vooraf uitgebreide 3D modellen van gebouwen gemaakt om ze daarna te herkennen en kunnen transformeren. Eens dergelijke gebouwen herkend worden door het navigatiesysteem (i.e. als men er naar kijkt), worden de 3D modellen getoond in plaats van de echte gebouwen. Hiermee kunnen de modellen aangepast worden in hoogte, breedte of kunnen ze zelf gebogen worden. De facade van het gebouw wordt op het 3D model geprojecteerd zodat de belichting overeenkomt met de realiteit.

Een dergelijke applicatie zou een uitbreiding kunnen zijn voor de Virtual LSD applicatie. In plaats van de visualisaties die in de Virtual LSD applicatie geïmplementeerd zijn, zouden 3D transformaties kunnen uitgevoerd worden op basis van de muziek.

2.5 Literatuurstudie rond immersie

Immersie is een abstract concept dat weergeeft in welke mate een gebruiker zich in een andere virtuele wereld waant. Dit concept is eerder slecht gedefinieerd en voor interpretatie vatbaar. In deze sectie worden verscheidene definities bekeken. Naast immersie bestaan er gelijkaardige concepten die meer duidelijk kunnen scheppen. Uit de bevindingen van de bestaande definities en gelijkaardige concepten wordt uiteindelijk een eigen definitie aan immersie gegeven, die zal gebruikt worden in de evaluatiesessies.

2.5.1 Duiding van immersie

Immersie kent verscheidene definities. Coomans[8] beschrijft immersie als *"the feeling of being deeply engaged where participants enter a make-believe world as if it is real."* Slater[50] benadrukt dat immersie een mapping nodig heeft tussen de proprioceptieve feedback (i.e. beweging van de spieren) van de gebruiker en de gegenereerde informatie in de virtuele wereld.

Immersie is een concept dat vaak gelinkt wordt aan andere concepten. Om een definitie aan immersie te kunnen geven, werd er gekeken naar bestaande concepten die sterk gelijk op immersie. De meest gebruikte concepten zijn Cognitive Absorption (CA), Flow en Presence. [22] We bespreken deze achtereenvolgens in volgende secties en geven daarna een eigen definitie aan immersie.

2.5.2 Flow

Flow (los vertaald als *in een mentale stroomversnelling zitten*) wordt beschreven als het proces waar de gebruiker een optimale beleving heeft. Csikszentmihalyi[10] verwoordde Flow als *"the state in which individuals are so involved in an activity that nothing else seems to matter"*. Wanneer gebruikers in de Flow status zitten, worden ze zo geabsorbeerd door hun activiteiten dat irrelevante gedachten en gebeurtenissen weggefilterd worden.

Flow wordt opgedeeld in 8 componenten[10]:

- Duidelijke doelen: mensen halen voldoening uit het najagen en volbrengen van doelen. Een voorbeeld hiervan zijn *quests* in World Of Warcraft.
- Directe feedback: feedback laat mensen weten dat ze een bepaald doel bereikt hebben of op de goede weg zitten. Hierdoor blijft het doel steeds voor ogen.
- Hoog concentratieniveau: een actie in de virtuele wereld vraagt zoveel concentratie dat slechts een beperkt deel van informatie nuttig lijkt. Irrelevante informatie zoals alledaagse bekommernissen worden doorgaans weggefilterd.
- Verlies van zelfbewustzijn: de informatie die men gebruikt om zichzelf voor te stellen lijkt niet relevant meer bij een actie of binnen virtuele wereld. Het laat toe om zichzelf te vergeten of grootser en beter voor te stellen.
- Verstoorde notie van tijd: na de Flow status weet men niet hoe de tijd verlopen is. De tijd die ervaren wordt in de virtuele omgeving lijkt niet dezelfde als die van de reële omgeving.
- Capaciteiten nodig hebben om uitdagende taak te volbrengen: door specifieke capaciteiten te gebruiken verhoogt het gevoel van macht, controle en zelfvoldoening. Bijvoorbeeld een vingervlugge hand-oog-coördinatie in een First Person Shooter laat toe om moeilijker taken te volbrengen.

- Sterke en moeiteloze betrokkenheid: men voelt zich betrokken zonder dat men zich hiervoor moet inspannen. Bijvoorbeeld een meeslepende film zorgt ervoor dat de kijker door beeld, verhaal, geluid en setting meevoelt met de fictieve personages, zonder dat de kijker zich moet inspannen om betrokken te zijn.
- Gevoel van persoonlijke controle: controle uitoefenen in een moeilijke situatie geeft het gevoel dat men zelf bepaalt hoe iets verloopt. Dit staat haaks tegenover het alledaagse leven waar men vaak controle verliest in situaties (e.g. een onverwachte vraag in een gesprek).

Flow lijkt zeer sterk op immersie te gelijken. Voornamelijk het verlies van zelfbewustzijn, de verstoorde notie van tijd en het hoge concentratieniveau zijn belangrijke componenten om immersie te beschrijven. Flow gaat echter vooral om optimale en daardoor extreme beleving. De vereisten om in een Flow status terecht te komen zijn hierdoor zeer strikt[22]. Brown[5] vond dat immersie geen extreem is, maar een graduele ervaring dat meerdere levels van betrokkenheid bevat.

2.5.3 Cognitive Absorption

Cognitive Absorption of Cognitieve Absorptie (CA) wordt omschreven als *“a state of deep involvement with software”*[1]. Gelijkaardig aan Flow, wordt CA opgedeeld in meerdere componenten. De 5 componenten (of dimensies) van CA zijn:

- Verstoorde notie van tijd: de tijd die ervaren wordt in de virtuele omgeving lijkt niet dezelfde als die van de rele omgeving.
- Hoog concentratieniveau: een actie in de virtuele wereld vraagt zoveel concentratie dat slechts een beperkt deel van informatie nuttig lijkt. Irrelevante informatie zoals alledaagse bekommernissen worden doorgaans weggefilterd.
- Toegenomen plezier: de interactie met de software (of VE) geeft een aangenaam en voldaan gevoel. Dit kan ontstaan door een doel te bereiken, een intrinsieke beloning of het gevoel dat men tot meer in staat is dan in de rele wereld.
- Controle: de perceptie van de gebruiker over de mate waarin hij/zij de situatie in handen heeft.
- Nieuwsgierigheid: de motivatie die voortvloeit uit de nieuwe beleving van de gebruiker. Nieuwe zaken willen ontdekken geeft een opwindend gevoel.

Ook bij CA zijn er enkele componenten die immersie lijken te beschrijven. De verstoorde notie van tijd, de verhoogde concentratie en het genot/plezier. Toch zijn beide begrippen niet helemaal gelijk. Het onderzoek in CA is vooral gefocust op het begrijpen en analyseren van gebruikersreacties op informatietechnologie, terwijl immersie eerder de beleving op een bepaald moment voorstelt[22].

2.5.4 Presence

Presence is een populair onderzoeksdomein vanaf de jaren 90. Er is echter heel wat onenigheid over de definitie van de term [60][50][58]. De definitie van Presence volgens Slater et al. [50] is *"the extent to which the actual system delivers a surrounding environment"*. Een vaak geciteerde definitie van Witmer en Singer [58] heeft het echter over *"the persons response to the VE system"*.

Presence is vooral gefocust op het meten en evalueren van Virtual Environments (VE). Er is dan ook veel onderzoek gegaan naar het ontwikkelen van questionnaires (en andere methodes) om Presence te kwantificeren. Witmer en Singer [58] richtten op een puur subjectieve meting van beleving. Zij vonden dat de mate waarin een interactie binnen de VE natuurlijk was, de Presence bepaalde. Zahorik en Jenison [60] keken meer naar de objectieve kant van de VE, namelijk het systeem. Zij concludeerden dat de mate waarin een actie van de gebruiker door het systeem als wettig (*"lawful"*) wordt beschouwd, de Presence bepaalt. Slater et al. [50] gebruikt eerder fysieke meetapparatuur om een directe link te vinden tussen de VE en de gebruiker. Hiermee benadrukt hij het nut van het virtuele lichaam binnen de VE.

Door deze zeer gerichte en praktische omschrijving, dat gericht is op VE, lijkt Presence enkele fundamentele componenten te missen om immersie te beschrijven.

2.5.5 Eigen definitie van immersie

Uit de 3 voorgaande termen kunnen we enkele gelijklopende kenmerken halen. Vooral de verstoorde notie van tijd, de verhoogde concentratie binnen de virtuele omgeving en het verlies van het zelfbewustzijn lijken een rol te spelen in immersie. Emotionele betrokkenheid is een andere belangrijke factor volgens Brown en Cairns [5].

Uit artistiek onderzoek [45][12] blijkt dat plezier en esthetiek een invloed hebben op de beleving/immersie. Als we deze bevindingen samen nemen, krijgen we 5 kerncomponenten die immersie beschrijven:

- *Attention*: De mate waarin de concentratie en de aandacht van de gebruiker naar de applicatie gaat. Irrelevante informatie zoals alledaagse bekommernissen worden doorgaans weggefilterd.
- *Enjoyment*: Plezier beleven met de applicatie. Alsook de mate waarin esthetiek (e.g. gebruik van gepaste vormen en kleuren) bijdraagt tot de beleving.
- *Emotional Attachment*: Gevoelens (e.g. blijdschap of angst) die vrijkomen door het gebruiken van de applicatie.
- *Transportation*: Het verlies van zelfbewustzijn en het geloof dat de virtuele wereld realistisch is.
- *Temporal dissociation*: De tijd die ervaren wordt in de virtuele omgeving lijkt niet dezelfde als die van de reële omgeving (Verstoorde notie van tijd).

In tegenstelling tot Jennet et al.[22] wordt immersie in deze masterproef niet gelimiteerd tot games. Elke interactie met een virtuele omgeving kan leiden tot immersie. In een AR applicatie, zoals Virtual LSD, kan de gebruiker even goed een immersieve ervaring hebben.

Hoofdstuk 3

Music Information Retrieval

Nadat we de basis hebben gelegd voor de applicatie, bespreken we in dit hoofdstuk het eerste onderzoeksdomein die voorkomt in de architectuur; Music Information Retrieval. MIR kan aanzien worden als het analyseren van muziek door er bepaalde informatie uit te halen. De literatuurstudie rond MIR zal om een basis vormen waarop de bespreking van de raamwerken en de implementatie steunt.

3.1 Literatuurstudie

3.1.1 Domein

In het domein van MIR spelen er verschillende facetten een belangrijke rol. Doordat er een aanzienlijk aantal facetten zijn waarmee men rekening moet houden in MIR research, is het verre van eenvoudig. Downie beschreef dit aspect van MIR als *The multifaceted challenge*[14]. We bespreken hier de verschillende facetten binnen MIR. Daarna worden enkele andere moeilijkheden toegelicht waarmee men rekening moet houden bij onderzoek in het domein.

De verschillende facetten binnen MIR zijn het *pitch*, *temporele*, *harmonische*, *timbrale* en *tekstuele* facet. Twee andere facetten die Downie bespreekt (*editoriaal* en *bibliografisch*) worden hier niet besproken. Het *editoriale* facet overlapt grotendeels met het *timbrale* facet. Het *bibliografische* facet beschrijft metadata, terwijl deze masterproef meer gericht is op de audiodata.

Pitch facet

Pitch of toonhoogte wordt aanzien als de waargenomen kwaliteit van het geluid. Deze is vooral afhankelijk van de frequentie (aantal oscillaties per seconde).

Temporele facet

Informatie dat afhankelijk is van de duur van een geluid, valt onder het temporele facet. De meest bekende voorbeelden hiervan zijn tempo en duur van een bepaalde toon. Het temporele facet geeft een muziekstuk een bepaald ritme.

Harmonische facet

Als er meerdere noten tegelijk gespeeld worden, ontstaat er een harmonie. Dit onderscheidt het ene muziekstuk van het ander. Door bepaalde noten op elkaar te laten volgen of samen te spelen met een bepaald tijdsinterval, krijgt een muziekstuk een harmonisch facet. Deze combinatie maakt een muziekstuk herkenbaar en creëert een toegevoegde waarde binnen muziek zoals wij ze waarnemen. Meerdere muzikanten die een muziekstuk spelen op een andere manier en/of instrument, hebben allemaal hetzelfde harmonische facet.

Timbrale facet

Het timbrale facet beschrijft het verschil in klankkleur. Een noot dat gespeeld wordt door een klarinet of een trompet geeft een zeer verschillend geluid. Elk instrument heeft een eigen timbre.

Ook de manier waarop een noot wordt aangeslagen of aangehouden, zit verwerkt in het timbrale facet.

Tekstuele facet

De lyrics (of songteksten) in een brede betekenis beslaan het tekstuele facet. Hymnes, koorteksten of libretti (teksten uit de opera) horen allemaal onder deze noemer. De tekst bij een muziekstuk is in een bepaalde mate onafhankelijk van de melodie en harmonie. Ze geeft echter een bepaalde boodschap en gevoel dat een muziekstuk lijkt te versterken. Desondanks de grote hoeveelheid informatie die uit lyrics kan gehaald worden, wordt dit facet niet gebruikt in deze masterproef. Dit laat ons toe om gebruik te maken van instrumentale muziek (e.g. klassieke muziek en filmmuziek).

Andere moeilijkheden

De *multifaceted challenge* is jammer genoeg niet het enige probleem binnen MIR onderzoek. Daarnaast is het analyseren van muziek uitdagend doordat muziek complex en subjectief is en kan gerepresenteerd worden in verscheidene manieren. Muziek kan voorgesteld worden in symbolen (e.g. partituren, muziknoten, tekst) of audio (e.g. live performance, cd, lp). Uit analyse blijkt ook dat muziek van nature uit een vorm van expressie is die anders gerealiseerd wordt afhankelijk van cultuur, en anders wordt ervaren door elk individu[14].

Iedereen heeft zijn eigen perceptie van muziek, waardoor het analyseren van de muzikale data een zeer moeilijk domein blijkt te zijn. Dit maakt het visualiseren van muziek in de omgeving van de gebruiker moeilijk. De Virtual LSD applicatie tracht bepaalde muzikale

facetten in de verf te zetten om de beleving te verhogen. Doordat iedereen een eigen perceptie heeft van muziek, is er sterke variatie mogelijk in de beleving van de gebruikers. Een gebruiker kan bijvoorbeeld gefocust zijn op de lyrics van de muziek, terwijl andere facetten (e.g. klankkleur, ritme) benadrukt worden door de Virtual LSD applicatie. De beleving van deze gebruiker kan dus sterk afwijken van andere gebruikers. In de evaluatie worden opmerkingen opgenomen die dit aantonen.

3.1.2 Basisoperatoren

Door middel van een reeks van veel gebruikte basisoperatoren kan men nuttige data uit muziek extraheren. Dankzij een combinatie van deze operatoren samen te voegen is het mogelijk om een audio feature (een representatie van bepaalde muzikale karakteristieken) te verkrijgen. In het verder verloop van deze masterproef worden deze basisoperatoren niet expliciet gebruikt. Er wordt gebruik gemaakt van MIR raamwerken die audio features extraheren. De basisoperatoren worden hier enkel beschreven om te schetsen hoe een audio feature geëxtraheerd wordt. Hiermee is het mogelijk om een eigen audio feature te ontwikkelen voor toekomstige systemen. We kunnen de basisoperatoren opsplitsen in de volgende categorieën; *transformaties*, *filters* en *aggregatiefuncties*[37][35].

Transformaties

Transformaties zijn operaties die data van een bepaalde vectorruimte naar een andere andere vectorruimte mappen. Een voorbeeld van een transformatie is de Discrete Fourier Transformatie (DFT). De DFT mapt data van het temporele domein naar het frequentiedomein en geeft de distributie van het signaal weer. Het temporele domein stelt het signaal voor dat verandert met de tijd. Het frequentiedomein toont de spectrale distributie van het signaal en laat het toe om de harmonie en toonaard te analyseren.

Filters

Filters zijn operaties die een reeks numerieke waarden naar een andere reeks numerieke waarden in dezelfde ruimte mappen. Een voorbeeld van een eenvoudige filter is de afgeleide van een reeks numerieke waarden. Een andere belangrijke type van filter is de *window* filter of vensterfunctie. Deze filter is een functie die nul is buiten een bepaald interval. Hierdoor lijkt het alsof men een stuk uit het volledige signaal knipt. Een veel gebruikte vensterfunctie in muziekanalyse is het Hamming-venster. Het Hamming-venster heeft een cosinusvorm waarbij een constante is bij opgeteld. Hierdoor is ze beter geschikt voor detectie van zwakkere componenten en ontstaan er minder discontinuïteiten aan de randen van het interval[34].

Aggregatiefuncties

Als laatste basisoperator hebben we de aggregatiefunctie, die een reeks van numerieke waarden mapt naar een enkele scalaire waarde. Een aggregatiefunctie dat vaak gebruikt werd in de Virtual LSD applicatie, is de standaard deviatie. Standaard deviatie is een statistische maat voor de spreiding van een variabele. Ze stelt een relatief compacte representatie voor van hoe het audiosignaal verandert over de tijd[34].

3.1.3 Audio features

Aan de hand van de beschreven basisoperatoren is het mogelijk om een wijde reeks van features te extraheren uit ruwe audiodata. Een verzameling van populaire audio features wordt opgesomd en besproken op basis van de bevindingen van Mckay[34] en Zelkowitz[37].

Features in het frequentiedomein

- **Spectral Centroid:**
Een benadering voor de helderheid van een muzieksignaal. Spectral Centroid is gedefinieerd als het zwaartepunt van het magnitude spectrum.
- **Spectral Roll-off Point:**
De frequentie onder een bepaalde fractie (gebruikelijk 0.85 of 0.95), van de distributie van het vermogenspectrum. Spectral Roll-off Point geeft een indicatie hoeveel energie van het signaal er in de lagere frequenties is. Deze feature wordt vaak gebruikt bij het onderscheiden van spraak in muziek.
- **Spectral Flux:**
Een maatstaf voor de hoeveelheid spectrale verandering in signaal voor ieder frame. Ze wordt vaak gebruikt als benadering voor het timbre.
- **Mel-Frequency Cepstral Coefficients (MFCC):**
Een meerdimensionale feature vector dat timbrale informatie voorstelt. De MFCC beschrijft de spectrale vorm van het signaal. De vector wordt berekend door de log-amplitude van het magnitude spectrum te nemen en te groeperen in bins op basis van de perceptueel gemotiveerde Mel-frequentieschaal. FCC is veel gebruikt in onderzoek rond spraak.
- **Linear Predictive Coding (LPC):**
Een methodologie dat is ontwikkeld om spraaksignalen te analyseren en te beschrijven. LPC tracht de basisparameters van het spraaksignaal te schatten. De basisparameters worden beschreven in formanten (spectrale banden die overeenkomen met resonantie-frequenties in het spraakkanaal) en de vocale transfer functie. LPC werd in het verleden gebruikt voor compressie en segmentatie van spraak.

Features in het temporale domein

- Zero Crossings:
Het aantal keer dat het signaal van positief naar negatief gaat en omgekeerd binnen een bepaalde tijdsperiode. Zero Crossings geeft een indicatie van de ruis in het signaal. Het is een populaire feature doordat het zeer eenvoudig en goedkoop te berekenen is.
- Root Mean Square (RMS):
Een maatstaf voor de gemiddelde energie in een signaal voor een bepaalde tijdsperiode. RMS komt vrij sterk overeen met het volume of geluidsterkte van een signaal.
- Beat Histogram:
Een compacte globale representatie van het ritme van een muziekstuk. Het beschrijft de herhalingsfrequentie van de hoofdbeat met de overeenkomstige sterkte.

3.1.4 Muziek, emotie en kleur

Muziek is een vorm om een bepaalde emotie te uiten[23]. Schubert[43] heeft aangetoond dat tempo, articulatie, ritme, pitch, harmonie en geluidsterkte een significante invloed hebben op de waargenomen emotie van muziek.

Psychologische studies [26][57] vonden dat emoties voortgebracht door muziek niet zo subjectief zijn dat ze ongeldig zijn voor wiskundige modellering. Bovendien werd aangetoond dat emotionele reacties op muziek binnen dezelfde cultuur zeer consistent zijn. Deze resultaten geven aan dat het gebruik en extraheren van emoties in MIR haalbaar is.

Het gebruik van emoties voortgebracht uit muziek laat ons toe om de muziekbeleving te vergroten. Omdat emotie een abstract niet-tastbaar begrip is, is er nood aan een concreet medium die een gevoel kan overbrengen. In 1986 speculeerde Arnheim[2] dat verschillende esthetische domeinen gerelateerd zouden zijn door gemeenschappelijke emotionele associaties. Concreet kunnen we dit zien als muziek die gelinkt is met kleur door emotie. Volgens Bresin[4] zorgt een interactieve display die de muziek voorstelt, voor een versnelling in het identificeren van de emotie. Door gebruik te maken van kleur en bewegende objecten wordt een bepaalde emotie sneller en mogelijk sterker ervaren. Er wordt dan ook verwacht dat de Virtual LSD gelijkaardige kenmerken toont.

Om een (in)directe link te vinden tussen muziek en kleur is er onderzoek gedaan in verscheidene subdomeinen. Een aantal methodes om symbolen (e.g. noten, toonladders) te mappen op emotie en kleur wordt beschreven door Cuddy[11].

Een andere aanpak is via het gebruik van ruwe audiodata. Door audio features te extraheren uit ruwe data, kan een emotionele classificatie worden opgemaakt. Recente studies [27][41] hebben aangetoond dat er duidelijke correlaties bestaan tussen audio features en emotionele categorieën. Zo heeft *Spectral Roughness* (i.e. een indicatie hoe regelmatig pieken voorkomen in het spectrum) een hoge correlatie met negatieve emoties zoals woede en angst. Een snel ritme of *onset rate* (aantal noten, beats, percussies in een seconde)

geeft een indicatie dat het muziekstuk vrolijk is. Geluidsterkte en volume toont een hoge correlatie met zeer uitgesproken emoties, zoals vrolijk en woedend. In de Virtual LSD applicatie werd de *Spectral Centroid* audio feature gebruikt om de kleuren lichter of donkerder te maken. Deze audio feature geeft een perceptuele helderheid van de muziek weer. In de visualisatie met bewegingsdetectie wordt de kleur bepaald door verscheidene audio features. Een hoge *Beat Sum*, die overeen komt met de *onset rate*, zal de kleuren meer tussen rood, paars en blauw weergeven. Als deze audio feature een lagere waarde heeft zullen meer positieve kleuren gekozen worden zoals groen en geel.

3.2 Raamwerken

Bij de meeste pogingen om muziekanalyse-systemen te bouwen eind 20ste eeuw, werd er getracht om de muziek te vertalen naar symbolen en daarna werd er muziektheorie gebruikt om de muziek te kunnen karakteriseren. Deze aanpak is aangekaart door Martin et al.[30] en Scheirer[42].

De huidige systemen proberen om structuur en de inhoud direct te analyseren van audio features, berekend op basis van het audiosignaal[55].

In de volgende deelsecties worden vijf Music Information Retrieval (MIR) raamwerken besproken. Daarna worden deze raamwerken met elkaar vergeleken op basis van praktische bruikbaarheid voor deze masterproef. Tenslotte lichten we de keuze voor het jMir raamwerk toe.

3.2.1 Marsyas

Marsyas (MusicAl Research SYstem for Analysis and Synthesis), gemaakt door George Tzanetakis, is een pionier in het gebied van *feature extraction*[33]. Het systeem bestaat uit een cliënt-server architectuur. De server, geschreven in C++, bevat alle signaalverwerking- en patroonherkenningsmodules die geoptimaliseerd zijn voor performantie. De cliënt, geschreven in Java, bevat enkel de gebruikersinterface code en communiceert met de server via sockets. Marsyas kan zowel gebruikt worden als algemeen classificatiesysteem als *feature extraction* systeem waar de uitkomst in het ARFF¹ formaat wordt beschreven[55]. Dit formaat kan hergebruikt worden in de Waikato Environment for Knowledge Analysis (WEKA) toolkit. Weka is een verzameling van machine learning algoritmen dat bestemd is voor data mining taken[18].

3.2.2 ALF

ALF (Audio-processing Library for Flash), ontwikkeld aan Drexel University, is een raamwerk dat algemene audioverwerkingsalgoritmes aanbiedt aan Flashontwikkelaars. ALF is

¹Attribute-Relation File Format, <http://weka.wikispaces.com/ARFF>

gebaseerd op Flash Player 10 en Adobe Alchemy, dat toelaat om C++ algoritmes te compileren naar Actionscript Virtual Machine om een significante verbetering in performantie te verkrijgen[13][36]. Het raamwerk is vooral gefocust op het gebruik bij real-time applicaties, vooral op muziek gebaseerde games (Pulse 2²).

Naast real-time uitvoer in Flash applicaties, is het ook mogelijk om CSV bestanden te genereren met de geëxtraheerde muzikale karakteristieken. Het raamwerk is aanspreekbaar via code of kan via een grafische gebruikersinterface gebruikt worden[44].

3.2.3 YAAFE

YAAFE (Yet Another Audio Feature Extractor), ontwikkeld in Institut Telecom de Paris-Tech, is een command line programma die het toelaat om muzikale features te extraheren. YAAFE deelt het verwerken van de audio in twee modules; *Parser* module en de *Engine* module. De eerste module, geschreven in Python, vertaalt het *feature extraction plan* naar computationele stappen in de tweede module. het *feature extraction plan* is een tekstbestand waar de gebruiker aangeeft welke audio features er moeten geëxtraheerd worden. De tweede module is geschreven in C++ voor het optimaliseren van de performantie. De uitvoer van het programma is een HDF5³ bestand. YAAFE is ontwikkeld om een groot aantal features gelijktijdig te extraheren uit een grote audio collectie[31].

Doordat er geen grafische gebruikersinterface aanwezig is, ligt de gebruiksvriendelijkheid van de applicatie lager.

3.2.4 MIRtoolbox

MIRtoolbox, ontwikkeld aan University of Jyväskylä, is een extensie bovenop Matlab dat gericht is op het extraheren van audio features uit een database van audiobestanden. De geëxtraheerde features kunnen gebruikt te worden in statistische analyse. Doordat MIRtoolbox bovenop Matlab werkt, is het mogelijk om zowel tekstbestanden te exporteren als grafieken te plotten binnen de Matlab-omgeving[25].

3.2.5 jMIR

jMir, ontwikkeld aan McGill University, is een softwaresuite gericht op MIR research. Het kan gebruikt worden om zowel audio als symbolische formaten (noten) te bestuderen, culturele informatie te verkrijgen van het web en een muziekcollectie te beheren[34].

Een onderdeel van het jMir raamwerk dat interessant leek te zijn voor deze masterproef is jAudio. jAudio is een raamwerk voor feature extractie dat gericht is op gebruiksvriendelijk en cross-platform gebruik. Het raamwerk is geschreven in Java, waardoor het op meerdere

²Pulse 2: Music-Reactive side scrolling game, <http://www.christianmhahn.com/pulse2>

³Hierarchical Data Format, <http://www.hdfgroup.org/HDF5/>

platformen kan gebruikt worden. Om het gebruiksgemak te verhogen is er een grafische gebruikersinterface en command line interface aanwezig, waaruit de gebruiker vrij kan kiezen. Er werd getracht om de algoritmes die nodig zijn voor audio extractie uitbreidbaar te maken, zodat ontwikkelaars hun eigen variaties of uitbreidingen op het raamwerk kunnen bouwen. jAudio laat toe de uitvoer van de audio feature extractie te exporteren als ACE⁴ XML of als ARFF formaat. Het ACE XML formaat laat toe om meerdimensionale audio features te groeperen als vectoren. In tegenstelling tot de meeste andere audio feature extractiesystemen[33][34].

3.2.6 Vergelijking

Er werd nagegaan welk raamwerk zou gebruikt worden voor de Virtual LSD applicatie. Om een oordeel te kunnen vellen werd enerzijds gekeken naar de mogelijke audio features die geëxtraheerd konden worden met behulp van de raamwerken. In tabel 3.1 zien we een overzicht van een reeks audio features die beschreven zijn per raamwerk. Als een raamwerk de mogelijkheid had om de desbetreffende audio feature te extraheren, zien we een kruisje in de overeenkomstige cel. Bij het ontbreken van deze functionaliteit, zien we een roodgekleurde cel.

We zien in tabel 3.1 dat het YAAFE raamwerk toch enkele essentiële features mist (die beschreven werden in 3.1.3). Zo ontbreekt een beat tracking functionaliteit en is het niet mogelijk om harmonische features te verkrijgen. Het ontbreken van de *Brightness* feature wordt genuanceerd doordat een combinatie van *MFCC* en *LPC* ook een benadering kunnen geven voor het timbre.

Tabel 3.1: Overzicht van audio features per raamwerk

		<i>Marsyas</i>	<i>ALF</i>	<i>JMIR</i>	<i>MIRtoolbox</i>	<i>YAAFE</i>
Tempo	Tempo - beat tracking	x	x	x	x	
	Beats per minute	x		x	x	
Speech	MFCC	x	x		x	x
	LPC	x	x	x		x
Timbre	Brightness	x	x	x	x	
Pitch	Zero Crossings	x		x	x	x
Power	Root Mean Square	x	x	x	x	x
Harmonic	Frequency	x	x	x	x	
	Amplitude				x	
Spectral	Rolloff	x	x	x	x	x
	Flux	x	x	x	x	x
	Centroid	x	x	x	x	x

In tabel 3.2 worden de raamwerken vergeleken op basis van algemene kenmerken. Aan de hand van deze kenmerken wordt de praktische bruikbaarheid van de raamwerken binnen

⁴Autonomous Classifier Engine, <http://weka.wikispaces.com/ARFF>

Tabel 3.2: Overzicht van muziekanalyse-raamwerken

	<i>Marsyas</i>	<i>ALF</i>	<i>jMIR</i>	<i>MIRtoolbox</i>	<i>YAAFE</i>
language	Java/C++	Actionscript/C++	Java	Matlab	Python/C++
interface	code/GUI	code/GUI	code/GUI/CLI	code/Matlab	command line
features	11	9	10	11	7
output	TXT/ARFF	CSV	ARFF, ACE	files, graphs	CSV, HF5
Playback functionality	x	x	x		
Export format parser			x	x	
real-time	x	x			
documentation	good	ok	ok	good	bad

deze masterproef geschat. Er werd niet gekeken naar performantie in deze vergelijking. De Virtual LSD applicatie maakt gebruik van audio feature extractie in een pre-processing stap. In tegenstelling tot real-time extractie/analyse, is de performantie van minder belang.

Programmeertaal

We zien een breed verschil in programmeertalen. Verscheidene raamwerken zijn geïmplementeerd in meerdere talen. Om performantieredenen werden de algoritmes vooral in C++ geschreven. Voor de gebruikersinterface daarentegen wordt voornamelijk Java (en Actionscript) gebruikt. Mijn programmeerachtergrond ligt bij Java en Actionscript, waardoor de desbetreffende raamwerken positief uitkomen.

Interface

We merken op dat enkel Marsyas, ALF en jMIR een grafische gebruikersinterface hebben. Daarnaast is er ook interactie via code mogelijk.

Features

Bij het onderdeel *features* worden de gegevens van tabel 3.1 gebruikt. Het getal geeft aan hoeveel features er kunnen geëxtraheerd worden door een raamwerk. Het maximum aantal features is twaalf. Zoals eerder opgemerkt scoort vooral YAAFE hier relatief slecht op.

Output

Hier worden de mogelijke exportformaten opgesomd. Deze variëren van zeer eenvoudige formaten, e.g. TXT en CSV, naar meer geavanceerde formaten, e.g. ARFF, ACE en HF5. Het ARFF-en HF5-formaat beschrijven gegevens in een zeer compacte wijze. Hierdoor blijft de bestandsgrootte klein, maar is het moeilijker leesbaar. Het ACE-formaat is een variatie op XML waardoor het minder complex is. Voor alle formaten zijn *parser*-programma's beschikbaar om het verwerken van de bestanden te vereenvoudigen. Enkel bij jMIR is

een dergelijk *parser*-programma geïntegreerd in het raamwerk. De andere raamwerken zijn afhankelijk van third-party programma's.

Playback functionaliteit

Enkel Marsyas, ALF en jMIR beschikken over functionaliteit om muziek af te spelen.

Exportformaat verwerkingsfunctionaliteit

Zoals eerder vermeld beschikt jMIR over een ingebouwde verwerkingsfunctionaliteit voor het ACE XML formaat. Ook MIRtoolbox beschikt over de functionaliteit om geëxporteerde bestanden opnieuw te verwerken door middel van het *mirgetdata(a)* commando.

Real-time interactie

Enkel Marsyas en ALF beschikken over interactie in real-time. Hier kunnen audio features berekend en opgevraagd worden terwijl de muziek speelt. Als een audiovisualisatie-applicatie gebruik zou maken van muziek die de gebruiker zelf kan kiezen, zijn deze twee raamwerken dus een zeer goeie keuze.

Documentatie

Tenslotte werd gekeken naar bestaande documentatie per raamwerk. De raamwerken Marsyas en MIRtoolbox hebben een uitgebreide documentatie, inclusief videofilmpljes en voorbeelden. Bij ALF en jMIR bestaat er enkel een functionele documentatie, zoals JAVADOC.

Gekozen raamwerk

Er werd beslist om voor het ontwikkelen van de Virtual LSD applicatie gebruik te maken van het jMIR raamwerk. Uit deze softwaresuite zal enkel jAudio gebruikt worden, dat instaat voor het extraheren van audio features. Deze keuze werd gemaakt omwille van de uitgebreide kennis van de Java programmeertaal en additionele mogelijkheden, zoals de ingebouwde playback-en parsingfunctionaliteit. De muziekanalyse van de Virtual LSD applicatie zou plaatsvinden in een preprocess-stap. Op deze manier wordt er CPU-rekentijd gespaard voor het computer visie- en visualisatiegedeelte. De zeer eenvoudige grafische interface van jAudio zorgt ervoor dat er op een intuïtieve manier audio features kunnen geëxtraheerd worden. Een waardig alternatief zou het Marsyas raamwerk zijn. Dit pioniersraamwerk scoort op alle opzichten goed. Het is meermaals gebruikt in commerciële toepassingen en heeft een zeer uitgebreide documentatie. Door het ontbreken van een ingebouwde verwerkingsfunctionaliteit, werd dit raamwerk niet gebruikt voor de Virtual LSD applicatie.

3.3 Implementatie

3.3.1 Audio features extraheren

Deze sectie zal de methodologie van het extraheren van audio features bespreken. Er wordt eerst een motivatie gegeven waarom deze stap niet real-time wordt uitgevoerd, maar eerder als *preprocessing*. Daarna wordt de grafische gebruikersinterface besproken van het jAudio raamwerk, die gebruikt werd om de audio features te extraheren. Tenslotte wordt iets meer gezegd over het exportformaat van dit raamwerk.

Preprocessing

Met behulp van het MIR raamwerk, jAudio, worden er audio features geëxtraheerd van een aantal liedjes. Deze extractie gebeurt als een *preprocess*-stap. De gebruiker van de applicatie komt in geen geval in contact met deze extractie. Het finaal prototype van de Virtual LSD applicatie biedt een beperkt aantal nummers aan, waar vooraf audio features van berekend werden. Hierdoor wordt er uitvoeringstijd bespaard tijdens het gebruik van de Virtual LSD applicatie. In tegenstelling tot het *computer vision*- en *computer graphics*-deel van de implementatie, is het mogelijk om deze berekeningen vooraf uit te voeren. Bij de andere delen van de implementatie is het absoluut noodzakelijk om real-time berekeningen uit te voeren, omdat ze afhankelijk zijn van de omgeving van de gebruiker. Voor het analyseren van muziek hebben we de omgeving van de gebruiker uiteraard niet nodig.

Doordat we deze stap vooraf uitvoeren, werd er niet gekeken naar performantie bij de raamwerken. Als men deze stap eerder wel real-time zou uitvoeren, zijn de raamwerken Marsyas en ALF, zoals besproken in 3.1.3, een betere keuze.

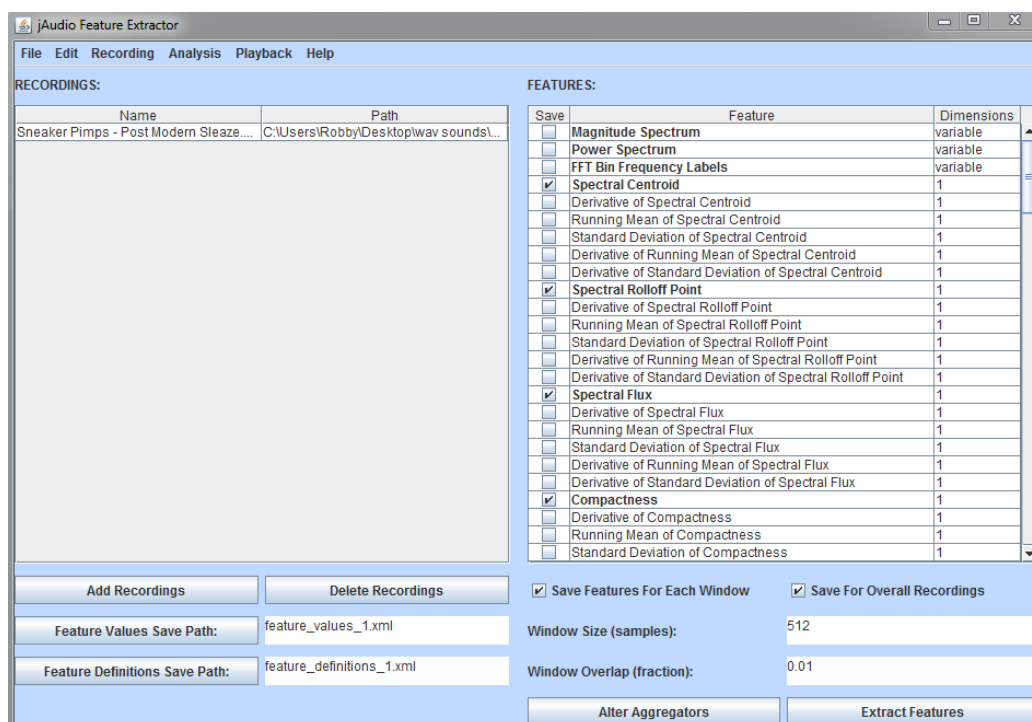
jAudio GUI

Voor het extraheren van audio features werd de grafische gebruikersinterface van jAudio gebruikt. In deze interface kan men een aantal liedjes kiezen om te analyseren. In afbeelding 3.1 zien we aan de linkerzijde een lijst van gekozen muzieknummers. Via de knop *add recordings* kunnen deze nummers geselecteerd en toegevoegd worden. Bij het extraheren van audio features voor deze masterproef doken er enkele problemen op bij het selecteren van de nummers. Zo gaf het merendeel van MP3 bestanden fouten bij het toevoegen aan de lijst. Bij verder onderzoek leek dit te komen doordat MP3 een third party formaat was in de tijd wanneer jAudio uitkwam. Hierdoor had de ontwikkelaar van het raamwerk geen recht om het MP3 formaat te analyseren. Ook leken liedjes van langere duur (> 3 minuten) problemen te geven met de Java omgeving. Hierbij werden er meerdere malen *Out Of Memory*-excepties gegooid. De ontwikkelaar van het raamwerk geeft toe dat grotere audiobestanden voor geheugenproblemen kunnen zorgen[34]. Deze problemen geven aan dat het raamwerk wel een update kan gebruiken. De laatste aanpassingen aan het raamwerk zijn gebeurd op 5 augustus 2010. Om deze problemen te omzeilen, werden de

liedjes die uitgekozen werden voor de Virtual LSD applicatie omgezet naar WAV⁵ formaat en ingekort tot 2 minuten.

Aan de rechterzijde van afbeelding 3.1, zien we een lijst van extraheerbare audio features. De audio features die in 3.1.3 werden besproken, komen voor in de lijst. Ook een aantal aggregatie-functies zijn bij elke audio feature mogelijk. Voor de Virtual LSD applicatie werd er enkel gebruik gemaakt van de aggregatie-functie; *standaard deviatie*. De standaard deviatie geeft een zeer gematigd verloop, in tegenstelling tot de meeste (ruwe) audio features die een sterk fluctuerend verloop tonen.

Onderaan rechts in afbeelding 3.1 kan men de instellingen voor het tijdsvenster kiezen. Het vinkje *Save Features For Each windows* geeft aan dat er een audio feature zal worden berekend voor elk tijdsvenster. Het tijdsvenster kan men kiezen door *Windows Size* in te geven. Een *Window Size* van 512 *samples* komt overeen met een tijdsvenster van 32 milliseconden. Door de *Window Overlap* te kiezen, zullen de tijdsvenster een fractie van elkaar overlappen. Dit zorgt ervoor dat er minder discontinuïteiten tussen 2 opeenvolgende tijdsvensters optreden.



Figuur 3.1: Grafische gebruikersinterface van jAudio

⁵Waveform Audio File Format, <http://www.kk.iij4u.or.jp/kondo/wave/mpidata.txt>

ACE XML exportbestand

De geselecteerde audio features worden per tijdsvenster weggeschreven naar een ACE XML bestand. De audio features worden beschreven in een *feature* tag, waar de naam en waarde terug te vinden zijn. De *feature* tags komen voor in meerdere *section* tags. De *section* tags stellen het tijdsvenster voor. Door een start- en stopattribuut kan nagegaan worden over welk tijdsvenster het gaat. In afbeelding 3.2 zien we de eerste regels van een ACE XML exportbestand, waar de *feature* en *section* tags in voorkomen.

Het exporteren van de audio features naar ACE XML brengt echter enkele moeilijkheden met zich mee. Doordat we audio features willen extraheren per tijdsvenster en een tijdsvenster slechts een klein tijdsinterval (e.g. 32 milliseconden) vertegenwoordigt, kan het bestand zeer groot worden. Elk ACE XML bestand dat in de Virtual LSD applicatie gebruikt wordt, telt meer dan 170000 regels. De grote bestanden zouden voor performantie-, geheugen-of implementatieproblemen kunnen zorgen bij een naïeve aanpak. Dankzij de ingebouwde *parser*-functionaliteit van jAudio, wordt dit echter zeer efficiënt behandeld. We behandelen het aanspreken van de audio features verder in sectie 3.3.3. Een andere moeilijkheid dat bij het exporteren naar ACE XML naar boven komt, is dat een bepaalde *feature* tag kan ontbreken als jAudio er niet in slaagt om de audio feature te extraheren. Vooral in het begin van een muzieknummer komt dit regelmatig voor. In afbeelding 3.2 zien we bijvoorbeeld dat in de eerste *section* tag slechts vier en in de tweede tag vijf audio features voorkomen. Ook hiervoor heeft de ingebouwde *parser*-functionaliteit van jAudio een oplossing. Het raamwerk laat toe om de *feature* tags te sorteren en te definiëren aan de hand van een lijst van audio features. Hiermee worden er *feature* tags aangemaakt met de waarde 0 (tijdens het lezen van het bestand) als deze zouden ontbreken. Concreet zal dus een *feature* tag aangemaakt worden in de eerste *section* tag van afbeelding 3.2 voor *Spectral Flux*, omdat die ontbreekt. Deze ingebouwde *parser*-functionaliteit is een niet te onderschatten voordeel. Dit benadrukt, zoals vermeld in 3.2, een argument om voor het jAudio raamwerk te kiezen.

3.3.2 Muziek afspelen

Voor het afspelen van de muzieknummers in de Virtual LSD applicatie werd eerst geopteerd voor de ingebouwde *playback*-functionaliteit van het jAudio raamwerk. Bij het ontwikkelen van het hulpsysteem waar audio features geplot werden in grafieken (verder besproken in 3.3.3), kwam er een groot probleem naar boven. Het was namelijk niet mogelijk om een accurate speeltijd (*current Media Time*) op te vragen aan de *Audio player* van jAudio. We hebben deze accurate speeltijd nodig omdat we real-time audio features willen gebruiken. Deze audio features zijn, zoals vermeld in 3.3.1, beschreven in zeer kleine tijdsintervallen van ongeveer 32 milliseconden. Om een audio feature te kunnen tonen op het juiste moment, moet de speeltijd van de *Audio Player* gebruikt worden om de overeenkomstige audio feature op te halen uit het ACE XML bestand. Na verder onderzoek naar alternatieven, werd er gekozen voor het JMF⁶ raamwerk. Het JMF raamwerk is een optionele uitbreiding

⁶Java Media Framework, <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html>

```

<?xml version="1.0"?>
<!DOCTYPE feature_vector_file>
- <feature_vector_file>
  <comments/>
  - <data_set>
    <data_set_id>C:\Users\Robby\Desktop\wav sounds\Triggerfinger - I Follow Rivers (Lykke Li cover).wav</data_set_id>
    - <section stop="0.0319375" start="0.0">
      - <feature>
        <name>Spectral Centroid</name>
        <v>0E0</v>
      </feature>
      - <feature>
        <name>Root Mean Square</name>
        <v>0E0</v>
      </feature>
      - <feature>
        <name>Zero Crossings</name>
        <v>0E0</v>
      </feature>
      - <feature>
        <name>Peak Based Spectral Smoothness</name>
        <v>0E0</v>
      </feature>
    </section>
    - <section stop="0.06075" start="0.0288125">
      - <feature>
        <name>Spectral Centroid</name>
        <v>0E0</v>
      </feature>
      - <feature>
        <name>Spectral Flux</name>
        <v>0E0</v>
      </feature>
      - <feature>
        <name>Root Mean Square</name>
        <v>0E0</v>
      </feature>
      - <feature>
        <name>Zero Crossings</name>
        <v>0E0</v>
      </feature>
      - <feature>
        <name>Peak Based Spectral Smoothness</name>
        <v>0E0</v>
      </feature>
    </section>
  </data_set>
</feature_vector_file>

```

Figuur 3.2: ACE XML exportbestand waarin de audio features beschreven worden

van de Java omgeving dat instaat voor het opnemen, afspelen, streamen en transcoderen van verschillende mediaformaten. In dit raamwerk is een *Player*-klasse aanwezig die het toelaat om de exacte speeltijd in (nano)seconden op te vragen. Het inladen en afspelen van een muziknummer vergt slechts 2 regels code. Eén voor de verwijzing naar het audiobe-stand en één voor het oproepen van de start methode. Een nadeel aan deze *Player*-klasse is dat het geen gebruik maakt van een aparte *thread*. De ingebouwde *playback*-functionaliteit van het *jAudio* had dit echter wel voorzien. Hierdoor kan het zijn dat de muziek stopt of hapert als de *Java Main Thread* problemen ondervindt. Dit probleem is echter zeer weinig voorgekomen tijdens het gebruik en evaluatie van de Virtual LSD applicatie.

3.3.3 Audio features aanspreken

Het moeilijkste deel van de audiogerelateerde implementatie was het real-time aanspreken van de geëxtraheerde audio features. Doordat we een klein tijdsvenster hanteren is het van groot belang dat de audio features accuraat worden opgevraagd.

De audio features worden binnen de Virtual LSD applicatie bijgehouden in een array die het ACE XML bestand voorstelt. Net zoals het ACE XML bestand is de array opgedeeld in secties (of tijdsvensters) met daarin een aantal audio features. Om de juiste sectie aan

```

645 private int getMusicFeature(AudioFeature feature){
646     if(getPlayer()!=null && loaded){
647         int index = (int) Math.ceil((getPlayer().getMediaTime().getSeconds() - 0.000748) / 0.032);
648         if(index > 0 && this.xml.sub_sets[index].feature_values.length > feature.index()){
649             double[] featureVal = this.xml.sub_sets[index].feature_values[feature.index()];
650             if(featureVal == null){
651                 return 0;
652             }
653             int limit = (int)(featureVal[0]*feature.factor());
654             return limit;
655         }else{
656             return 0;
657         }
658     }else{
659         return 0;
660     }
661 }

```

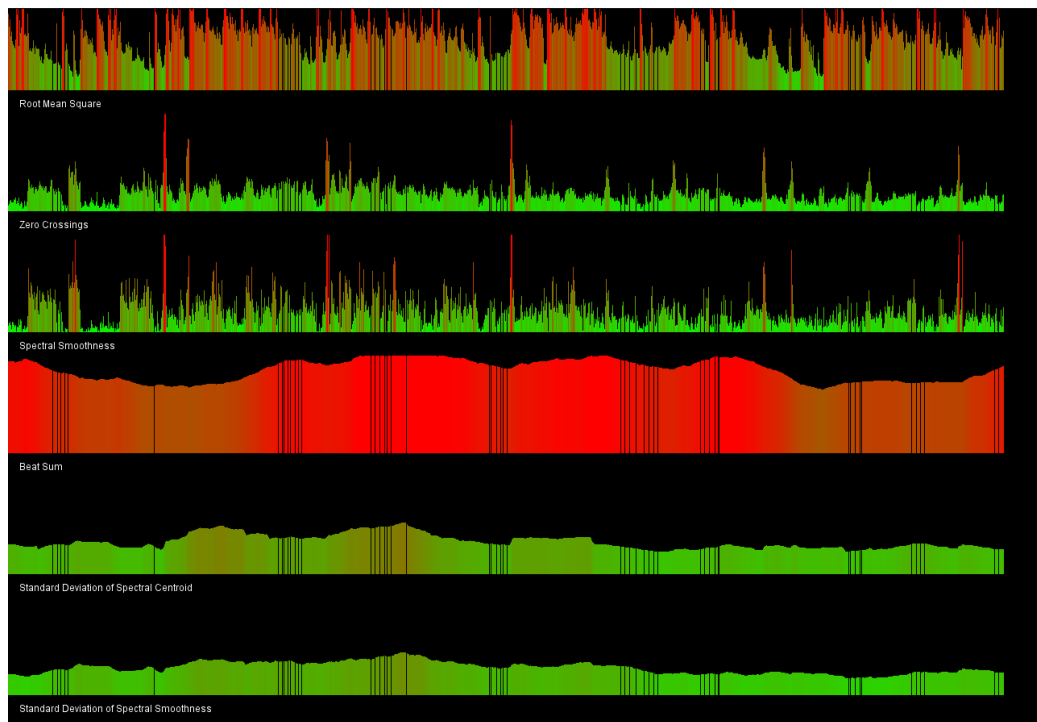
Codevoorbeeld 3.3: Algoritme om real-time audio features op te vragen

te spreken op een bepaald moment in de Virtual LSD applicatie, hebben we een *integer* nodig die de *index* voorstelt in de array. In codevoorbeeld 3.3 zien we op regel 648 de *index* variabele die een plaats in de *xml* array voorstelt. Deze index wordt berekend op basis van de exacte mediatijd gedeeld door de duur van een tijdsvenster (e.g. 32 milliseconden). De berekening van de index variabele staat beschreven op regel 647 in codevoorbeeld 3.3.

Om een nuttige keuze te maken van audio features, werd er naast het bestuderen van de literatuur ook een systeem ontwikkeld om deze te plotten in een grafiek. Door meerdere onder elkaar te plotten, konden gelijkaardige audio features geëlimineerd worden. We willen immers audio features in de Virtual LSD applicatie gebruiken die verschillende aspecten van de muziek representeren. Om deze proper in een grafiek te krijgen, moesten de waardes genormaliseerd worden. Een audio feature zoals *Root Mean Square* varieert tussen 0 en 4×10^{-1} , terwijl *Spectral Flux* varieert tussen 0 en 17×10^{-3} . Om op een nuttige manier gebruik te kunnen maken van de audio features, werden alle waardes herleid naar het bereik [0,100]. Voor elke audio feature werd een factor bepaald, waarmee de waardes werden gewogen. Deze normalisatie laat ons toe om alle audio features te plotten in grafieken met hetzelfde assenstelsel. Daarenboven zijn de genormaliseerde audio features eenvoudiger te gebruiken in de Virtual LSD applicatie. Door het bereik [0,100] te gebruiken, kan men een visualisatie wegen met een percent.

In afbeelding 3.4 zien we 6 audio features onder elkaar geplott in een grafiek. De (horizontale) x-as stelt de tijd voor in milliseconden, waarbij 1 pixel 32 milliseconden vertegenwoordigt. De y-as stelt de waarde van de audio feature voor in het overeenkomstig tijdsvenster. De waardes variëren tussen 0 en 100 en worden getoond als gekleurde verticale lijnen. De kleur varieert tussen groen (bij waarde 0) en rood (bij waarde 100). Het systeem werd zo ontwikkeld dat de audio features real-time getekend werden in grafieken. Elke 32 milliseconden wordt er een lijn getekend die de waarde van de audio feature in het overeenkomstig venster voorstelt. Op deze manier konden de audio features vergeleken en gecontroleerd worden met het muziknummer dat tezelfdertijd afspeelde.

Om in de Virtual LSD applicatie gebruik te maken van de geëxtraheerde audio features, werden alle visualisaties beschreven in een Enumeratie-klasse *AudioFeature*. In deze klasse



Figuur 3.4: Audio features geplot in grafieken. De waarde van een audio feature wordt getekend langs de y-as. De x-as geeft het tijdsverloop aan.

staan alle audio features opgelijst en wordt een *index* en *factor* bijgehouden. De *index* stelt de plaats in de ACE XML array voor (binnen een tijdsvenster). De *factor* representeert de eerder besproken normalisatiefactor. In codevoorbeeld 3.3 zien we dat de methode *getMusicFeature* een *AudioFeature* object als argument verwacht. Binnen deze methode wordt ook gebruik gemaakt van de *index* en *factor*.

Hoofdstuk 4

Computer visie

Computer visie (CV) is een domein waar algoritmes en methodes worden ontwikkeld om data uit de echte wereld om te zetten naar een numerieke of symbolische vorm, zodat het kan gebruikt worden door een computer. In deze masterproef zal de omgeving van de gebruiker moeten omgezet worden in bruikbare data, zodat een gepaste visualisatie kan getoond worden. Zo kunnen CV-algoritmes informatie halen uit de webcam-beelden van de applicatie om iets meer te kunnen zeggen over de omgeving. Opnieuw beginnen we hier met een literatuurstudie omtrent CV om daarna de raamwerken en de implementatie te kunnen bespreken.

4.1 Literatuurstudie

Computer visie kent meerdere onderzoeksgebieden. Er werden slechts enkele bestudeerd om informatie uit het webcam-beeld te kunnen halen. Alhoewel veel vooruitgang is geboekt in CV-onderzoek, zijn veel technieken en toepassingen afhankelijk van een gecontroleerde omgeving. De Virtual LSD applicatie gebruikt echter geen gecontroleerde omgeving. Ze wordt gebruikt op meerdere plaatsen (zowel binnen als buiten) waar de belichting niet gecontroleerd werd. Hiermee konden de gebruikers een realistische indruk krijgen van een toekomstige commerciële applicatie. Deze ongecontroleerde omgeving brengt echter heel wat moeilijkheden met zich mee. In deze sectie bespreken we enkele klassieke CV-technieken die meer informatie kunnen geven uit een afbeelding. In de Virtual LSD applicatie kunnen deze technieken gebruikt worden om informatie uit de webcam-beelden te halen.

4.1.1 Detecteren van *interest points*

Een *interest point* is een plaats in een beeld dat interessante eigenschappen heeft. De eigenschappen (e.g. intensiteit, kleur en textuur) van een *interest point* verschillen doorgaans sterk van zijn directe omgeving. Een *interest point* kan een punt zijn in een beeld, maar ook een hoek (of contour) van een object in het beeld[54]. Onderzoek [3] heeft aangetoond dat hoeken essentieel zijn bij de mens om objecten en scènes te herkennen. Plaatsen in het

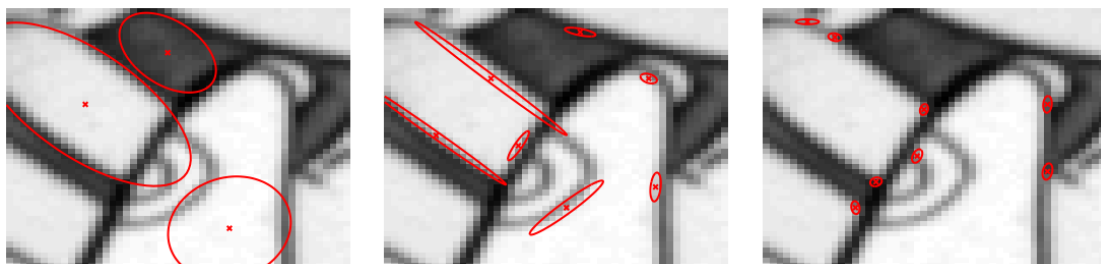
beeld die deze hoeken vertegenwoordigen zijn doorgaans een *interest point*.

Er zijn verschillende technieken om *interest points* te detecteren in een beeld. Veel van deze technieken focussen zich op het detecteren van een sterke verandering in één of meerdere eigenschappen. Voorbeelden hiervan zijn intensiteit-gebaseerde detectie en methodes die kleurinformatie exploiteren[54]. We bespreken hier enkel de meest bekende methodologie, namelijk intensiteit-gebaseerde detectie. Andere (complexere) methodologieën worden hier niet opgenomen omdat dit buiten het bereik van deze masterproef zou vallen.

Intensiteit-gebaseerde detectie

Veel intensiteit-gebaseerde methodes maken gebruik van eerste en tweede orde afgeleiden van een zwart-wit beeld. Het meest bekende *interest point* detectie-algoritme is de *Harris corner detector*. Dit algoritme zoekt punten die lokaal een maximale variantie bereiken in een bepaalde richting. Een gebied in het beeld (e.g. 3 op 3 pixels) wordt enkele pixels verschoven in verscheidene richtingen om de gemiddelde verandering in intensiteit te bepalen. De verandering in intensiteit wordt bepaald door eerste orde afgeleiden van de naburige pixels te berekenen en in een *second moment* matrix te steken[54]. Aan de hand van de eigenwaarden van deze matrix kunnen we hoeken, randen of vlakke regio's herkennen. Een regio wordt bepaald door de twee eigenwaarden van de matrix. In figuur 4.1 zien we dat in de meest linkse afbeelding vlakke regio's herkend worden, doordat de eigenwaarden in de *second moment* matrix beide klein zijn. In de middelste afbeelding zien we randen, doordat er een grote en een kleine eigenwaarde is. In de meest rechtse afbeelding zien we hoeken doordat beide eigenwaarden groot zijn. In figuur 4.2 zien we welke punten het algoritme detecteert in een beeld. In de rechtse afbeelding stellen de witte punten gedetecteerde hoeken voor.

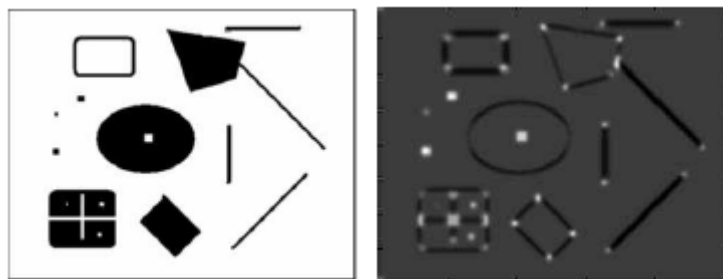
Figuur 4.1: Regio's die gevonden kunnen worden door Harris Corner detectie-algoritme. Er kunnen vlakke regio's (links), randen (midden) en hoekpunten (rechts) herkend worden. Bron: Czech Technical Univeristy Prague



4.1.2 Detecteren van contouren

Zoals eerder vermeld, zijn de contouren en hoeken van objecten essentieel om objecten en scènes te herkennen. In deze sectie bespreken we kort het detecteren van contouren (*edge*

Figuur 4.2: Het Harris Corner detectie-algoritme toegepast op een beeld (links) en de gedetecteerde hoeken (rechts).

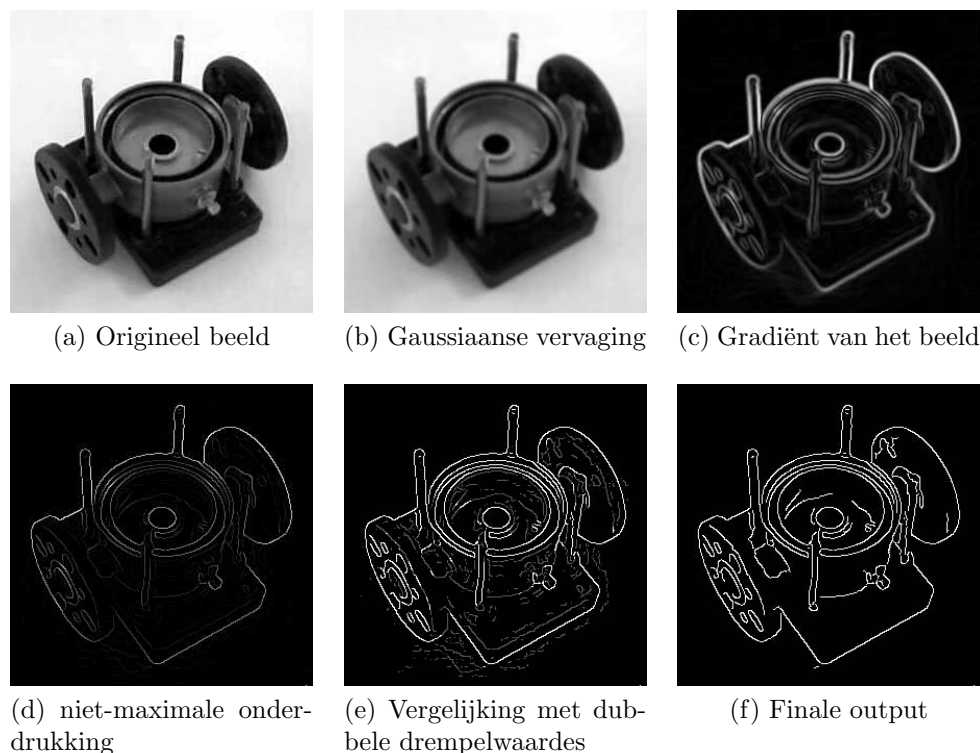


detection) en geven we een bekend algoritme (Het Harris Corner detectie-algoritme), dat gebruikt werd in de Virtual LSD applicatie.

Voor het detecteren van contouren (of segmenten omsloten door contouren) zijn er twee verschillende strategieën. De eerste is het detecteren van de contouren van de objecten door randen te vinden in een afbeelding en deze te linken aan andere gevonden randen in het beeld. De tweede strategie zoekt naar regio's in het beeld met gelijkaardige eigenschappen en laat deze groeien tot een discontinuïteit gevonden wordt. Een discontinuïteit betekent hier een sterke verandering in een eigenschap (e.g. helderheid). Een dergelijke verandering komt meestal overeen met een verandering in diepte, richting, textuur of belichting in de scène[29]. We bespreken hier de eerste strategie verder aan de hand het meest bekende[59] algoritme voor het detecteren van contouren, het *Canny* detectie-algoritme[7]. Dit algoritme werd gekozen omdat dit het meest voor de hand liggende is en duidelijk te begrijpen valt.

***Canny* detectie-algoritme**

Het *Canny* detectie-algoritme[7] is een algoritme dat in meerdere fases verloopt. In een eerste fase wordt de ruis in het beeld verminderd door een Gaussiaanse vervaging toe te passen (figuur 4.3b). Ruis in het beeld zorgt meestal voor discontinuïteiten in helderheid of kleur, waardoor dit mogelijk als contour kan beschouwd worden. Een Gaussiaanse vervaging is een filter dat pixels meer zal vervagen naarmate ze verder van het midden van het beeld liggen. In een tweede fase wordt de gradiënt van het beeld berekend door een *Sobel* operator toe te passen (figuur 4.3c). Hier zullen eerste-orde afgeleiden berekend worden in verscheidene richtingen (e.g. horizontaal, verticaal en soms diagonaal). Deze stap benadrukt de contouren van objecten zeer duidelijk, maar de contouren zijn nogal breed. We wensen een dunnere contour om meer precisie te hebben, waardoor randen beter aansluiten bij de realiteit. Dit wordt verwezenlijkt in de derde fase van het algoritme, de niet-maximale onderdrukking. In deze fase wordt binnen een (brede) contour enkel de maximale waardes overgehouden (figuur 4.3d). In een vierde en laatste fase wordt het beeld vergeleken met twee drempelwaardes. Contouren met een waarde boven de hoogste drempelwaarde, worden als sterke contour beschouwd. In figuur 4.3e worden deze getekend als witte lijnen. De contouren die tussen de laagste en hoogste drempelwaarde schommelen,



Figuur 4.3: De meerdere fases van het *Canny* detectie-algoritme. Bron: Moeslund[38]

worden als zwakke contour beschouwd. Deze worden in figuur 4.3e in grijs getekend. Sterke contouren behoren tot de finale output, zwakke contouren worden enkel aan de finale output toegevoegd als ze direct verbonden zijn met een sterke contour. De verschillende fases en de finale output van het *Canny* detectie-algoritme zijn te zien in figuur 4.3.

4.1.3 Toepassingen

Er zijn veel toepassingen die gebruik maken van *interest points* en contouren. De toepassingen die gerelateerd zijn aan deze masterproef zijn objectherkenning en *tracking*. We bespreken deze kort in de komende secties en geven aan hoe ze zouden kunnen gebruikt worden in de Virtual LSD applicatie. Onderstaande toepassingen kunnen mogelijke uitbreidingen of alternatieve benaderingen zijn.

Objectherkenning

Objectherkenning is een populaire toepassing dat kan verdeeld worden in twee types. De eerste is het herkennen van specifieke objecten. Een bestaand object (e.g. koffietas met KULeuven logo) wordt bijvoorbeeld gezocht in een foto. Het tweede is het herkennen van een categorie van objecten. Hier wordt er in een beeld gezocht naar objecten die passen bij een categorie (e.g. auto). Dit is een duidelijk moeilijkere taak, doordat niet alle objecten

binnen een categorie (e.g. auto's) een identiek voorkomen hebben. Voor beide types van objectherkenning worden *interest points* gebruikt. Er worden *interest points* gehaald uit een foto en vergeleken met een specifiek object of een database van objecten[54].

In de Virtual LSD applicatie zou objectherkenning van pas komen om specifieke objecten uit het webcam-beeld te halen. Zo zouden auto's, bloemen of wandelende mensen kunnen herkend worden. Deze objecten zouden dan kunnen gebruikt worden om visualisaties te genereren. Zo zouden visualisaties de informatie exploiteren van het type object dat herkend werd. Om objectcategoriën te herkennen zou een database opgesteld moeten worden waar duizenden objecten binnen die categorie zijn opgeslagen. Dergelijke databases zijn vrij beschikbaar¹. Maar herkenningsalgoritmen ontwikkelen die performant en robuust genoeg zijn om real-time te kunnen gebruiken, zou ons te ver leiden binnen deze masterproef. De ontwikkeling op zich zou een te groot aandeel van deze masterproef opnemen en meer onderzoek zou nodig zijn. Een dergelijk systeem werd beschreven door Zu et al.[53]. Hierin wordt tekstueel aangegeven wat er op het beeld te zien is (e.g. "Rode auto stopt voor rood licht. Groene auto slaat rechts af"). Dit systeem maakt echter gebruik van een camera die stilstaat, waardoor *background subtraction* kan worden toegepast. Enkel beelden die niet in de achtergrond voorkomen, zullen geanalyseerd worden.

Tracking

Als objectherkenning zou gebruikt worden in de Virtual LSD applicatie, zou het zeer CPU-intensief zijn om in elk frame de objecten te herkennen. Eens een object zou herkend zijn, zou men een andere techniek kunnen gebruiken om het object te volgen in de komende frames. Deze techniek noemt men *tracking*. Door een object te volgen kan er extra informatie uit het beeld gehaald worden. *Tracking* houdt namelijk informatie bij over meerdere frames [16]. Hiermee zou bijvoorbeeld de richting van een voorbijrijdende auto kunnen achterhaald worden. Deze richting zou dan kunnen gebruikt worden om visualisaties te genereren in het kielzog van de auto of net in de richting dat de auto rijdt. Omdat objectherkenning te complex bleek voor binnen deze masterproef, werd *tracking* dan ook niet gebruikt in de Virtual LSD applicatie.

4.2 Raamwerken

Om de omgeving van een Virtual LSD gebruiker te verstaan, werd er naar relevante informatie gezocht in een camerafeed (die het zicht van de gebruiker voorstelt). Om deze informatie te zoeken werden in deze masterproef twee open source computer visie raamwerken onderzocht. Het populaire OpenCV en het recent uitgebrachte (november 2011) BoofCV. Deze sectie zal beide raamwerken vergelijken, beschrijven en aantonen wat de mogelijkheden ervan zijn. We beginnen met OpenCV en zullen daarna BoofCV beschrijven.

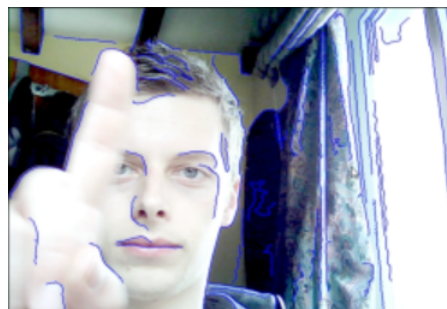
¹LHI database, <http://www.imageparsing.com/FreeDataOutline.html>

4.2.1 OpenCV

OpenCV² is het meest gebruikte open source CV raamwerk dat beschikbaar is. Het is volledig geschreven in C en was de eerste jaren van zijn bestaan ook enkel aanspreekbaar in C. Ondertussen bestaan er een aanzienlijk aantal *wrapper*-klassen die het mogelijk maken om dit raamwerk aanspreekbaar te maken in andere programmeertalen. Een voorbeeld hiervan is JavaCV³, die het mogelijk maakt om via Java de in C geschreven algoritmes aan te spreken.

Om dit raamwerk te testen werd de *wrapper*-klasse JavaCV gebruikt. Er werd een test-project gemaakt dat contouren uit het webcam-beeld haalt en tekent op het scherm. Elke methode die gebruikt maakt van OpenCV algoritmes wordt in JavaCV aangesproken met prefix "cv". Het *Canny* detectie-algoritme wordt bijvoorbeeld toegepast door de *cvCanny*-methode. Alle methodes kunnen hierdoor opgezocht worden in de uitgebreide documentatie van OpenCV. Alhoewel JavaCV toelaat om in de Java omgeving te programmeren, is de manier waarop moet geprogrammeerd worden niet zoals in Java. De syntax is dus volledig Java-gebaseerd, maar men moet nog steeds geheugenafhandeling zelf implementeren zoals in C. Dit zorgt ervoor dat de objecten (e.g. webcam-beeld of contouren) wijzers zijn naar het geheugen in plaats van datastructuren. De ingebakken methodes (met prefix "cv") gebruiken deze objecten zonder problemen, maar als men specifieke data uit een object wil halen om te visualiseren, wordt het moeilijk. Een concreet voorbeeld is het object dat de contouren voorstelt na het toepassen van het *Canny* detectie-algoritme. Men kan deze contouren tekenen op het scherm met de *cvDrawContours*-methode. Maar de punten in de contour overlopen om een eigen visualisatie (zie 5.3.3) te maken, bleek quasi onmogelijk. In figuur 4.4 zien we de contouren die getekend worden met de *cvDrawContours*-methode.

Figuur 4.4: Contouren die gedetecteerd en gevisualiseerd worden met het OpenCV-raamwerk



²OpenCV, <http://opencv.willowgarage.com/wiki/>

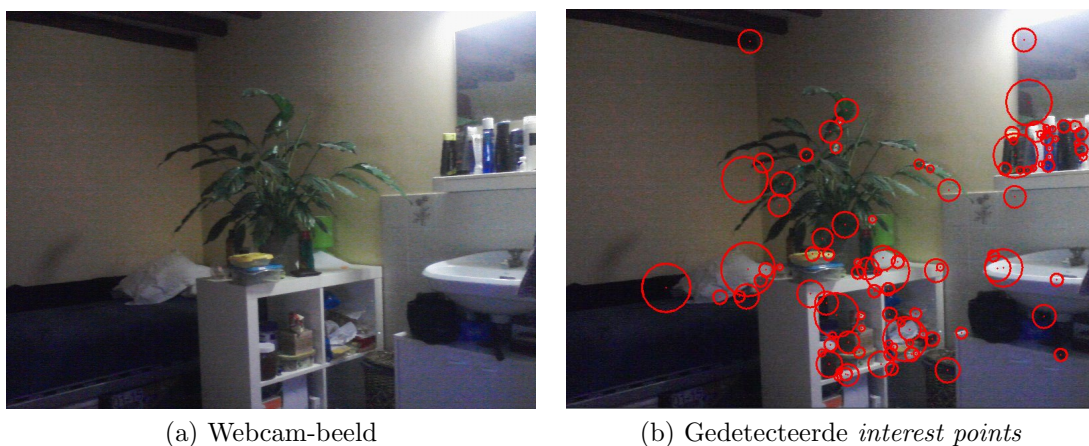
³JavaCV, <http://code.google.com/p/javacv/>

4.2.2 BoofCV

BoofCV⁴ is een gloednieuw open source CV raamwerk dat vooral gefocust is op real-time computer visie applicaties. Dit raamwerk is volledig geschreven in Java, waarvan de leercurve minder steil is dan de C programmeertaal. De *alpha release* (v0.1) is pas vrijgegeven in november 2011. Hierdoor konden de makers van dit raamwerk de laatste nieuwe algoritmes in CV gebruiken. Door de jonge aard kampt dit raamwerk aan de andere kant met bugs, waardoor het minder betrouwbaar is. Ondertussen is versie alpha v0.8 vrijgegeven met bijkomende functionaliteit en een verbetering in betrouwbaarheid.

Het raamwerk is zeer toegankelijk door de talloze *online applets* die beschikbaar zijn op de website van BoofCV. Deze toepassingen laten toe om functionaliteit van het raamwerk uit te testen. Daarenboven zijn er veel code-voorbeelden aanwezig die stapsgewijs een methode of algoritme uit de doeken doen.

Er werd besloten om dit nieuwe raamwerk te testen door enkele functionaliteiten op de proef te stellen in een eigen project. Het detecteren van *interest points* (eerder besproken in 4.1.1) is één van deze functionaliteiten. In figuur 4.5 wordt hiervan het resultaat getoond. Het detectie-algoritme produceert een lijst van punten die vrij te gebruiken zijn. Hiermee kunnen de *interest points* direct getekend worden (zoals in dit voorbeeld) of kunnen deze gebruikt worden om andere visualisaties te creëren.



Figuur 4.5: Het detecteren van *interest points* met het BoofCV-raamwerk.

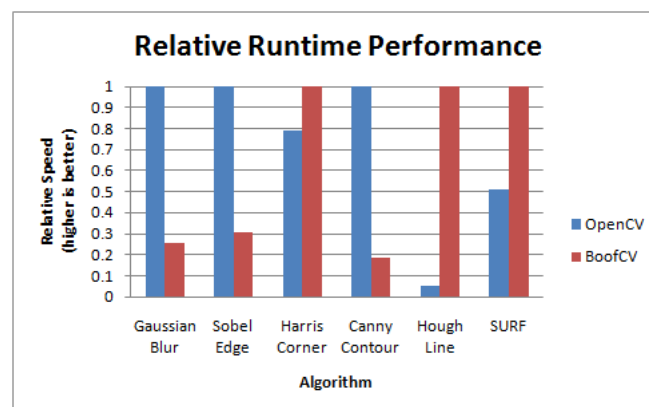
Naast het detecteren van *interest points* werd ook een test uitgevoerd om contouren te detecteren, gelijkaardig aan de test met OpenCV. Het vergelijken van deze testen gaf wel enkele verschillen. OpenCV bleek wel degelijk een hogere performantie te hebben. De framerate bij de tests was bij OpenCV hoger, waardoor de beelden vlotter oogden. Bij de test met BoofCV werd een extra raamwerk gebruikt om de webcam-beelden op te halen. Dit raamwerk wordt hierna besproken en werd in het tweede prototype veranderd om een

⁴BoofCV, <http://www.boofcv.org>

verhoging van performantie te verkrijgen. De lagere framerate van de BoofCV test is dus deels hieraan te wijten.

De ontwikkelaar van het BoofCV-raamwerk heeft een vergelijkende studie uitgevoerd met OpenCV. In figuur 4.6 zien we een vergelijking in uitvoeringssnelheid voor bepaalde algoritmes. De snelheden zijn relatief uitgedrukt, waardoor een hogere score beter is. Deze grafiek bevestigt de bevindingen uit de *Canny* detectietest, dat OpenCV hier sneller presteert. Het detecteren van *interest points* (*Harris*, *Hough Line* en *SURF*) is dan weer sneller met BoofCV. Om redenen van performantie kunnen we dus geen duidelijke keuze maken tussen BoofCV en OpenCV. Het ontbreken van bruikbare datastructuren in JavaCV (door de onduidelijke geheugenverwijzingen) bleek echter de implementatie aanzienlijk moeilijker te maken. Om deze reden werd dan ook beslist om te kiezen voor het BoofCV-raamwerk.

Figuur 4.6: Vergelijkende studie tussen BoofCV en OpenCV. OpenCV scoort beter op *low level* algoritmes, terwijl Boofcv beter presteert *high level* operaties zoals het detecteren van *interest points*. De snelheden zijn relatief uitgedrukt, waardoor een hogere score beter is. Bron: www.boofcv.org



4.3 Implementatie

In deze sectie wordt de implementatie van het computer visie gedeelte van de Virtual LSD applicatie besproken. Alle CV-gerelateerde zaken worden afgehandeld in een aparte *Thread*. Een *Thread* is een deel van een proces en laat toe om onderdelen gelijktijdig uit te voeren. Hierdoor blijft de visualisatie-*Thread* lopen zelfs als er iets fout loopt in een CV algoritme. Daarenboven maakt het de Virtual LSD applicatie transparant en makkelijk aanpasbaar. In de komende secties wordt uitgelegd hoe de webcam-beelden werden opgehaald en welke informatie eruit gehaald werd.

4.3.1 Webcamfeed ophalen

Het webcam-beeld is de enige data dat geanalyseerd wordt in de Virtual LSD applicatie om de omgeving van de gebruiker te 'begrijpen'. Het is dus niet voldoende om enkel de

webcam-beelden te tonen, maar deze moeten opgeslagen worden in het geheugen om te kunnen analyseren. De standaard Java omgeving laat het echter niet toe om beelden vanuit een webcam op te halen. Processing biedt wel webcam-functionaliteit aan, maar deze is afhankelijk van Quicktime⁵. Bovendien gaf deze functionaliteit onstabiele resultaten. Er werd dus maar naar alternatieven gezocht. In het eerste prototype werd gekozen voor het Java Media Framework, dat eerder werd besproken in 3.3.2. Na de eerste evaluatiesessie bleek dat de webcam-beelden niet responsief genoeg waren. Dit was nefast voor de beleving van de gebruikers. In het tweede prototype werd GSVideo gebruikt, een raamwerk dat bovenop Processing draait. De performantie van beide raamwerken werd vergeleken en GSVideo bleek aanzienlijk betere resultaten te geven. We bespreken hier de implementaties uit de twee prototypes van de Virtual LSD applicatie.

Eerste prototype (JMF)

In het eerste prototype maakte de Virtual LSD applicatie gebruik van het JMF raamwerk om muziek af te spelen. Dit raamwerk voorziet ook functionaliteit om een webcam aan te spreken. Voor een eerste prototype leek dit een goeie optie om deze functionaliteit dan ook te gebruiken. In codevoorbeeld 4.7 zien we een codevoorbeeld om de webcam-beelden op te halen en te tekenen op het scherm. De *loadCam*-methode (regels 47-54) initialiseert het *Player*-object dat de webcam voorstelt. Om connectie te kunnen maken met de webcam, moet de apparaatnaam gekend zijn (regels 48-49). Als men connectie heeft gemaakt met de webcam kunnen de compatibele videoformaten opgevraagd en gekozen worden. In de Virtual LSD applicatie wordt gebruik gemaakt van het RGB videoformaat waarbij de resolutie van de webcam-beelden 640 op 480 pixels zijn.

De webcam-beelden kunnen op twee verschillende manieren getekend worden. Er is een mogelijkheid om de beelden te laten tekenen door het JMF raamwerk. Hierbij zal een nieuw venster geopend worden waar de webcam-beelden in zullen getekend worden. We kunnen hier echter niet aan de data in de webcam-beelden. Doordat we informatie uit die beelden willen halen, werd deze mogelijkheid weerhouden. Een tweede mogelijkheid is elk webcam-beeld ophalen met de *FrameGrabbingControl*(regels 60-61). Deze ietwat omslachtige methode maakt een *Buffer*-object aan waar het webcam-beeld als pixels worden opgeslagen. Deze buffer kan omgevormd worden naar de *Image*-datastructuur van de Java omgeving. Hieruit kan informatie gehaald worden of het *Image*-object kan ook getekend worden met Processing. Uit de eerste evaluatiesessie bleek dat de camera-beelden niet nauwkeurig waren en dat dit een storende factor was voor de applicatie. Voor het tweede prototype werd gezocht naar een alternatief voor het ophalen van de webcam-beelden.

Tweede prototype (GSVideo)

Voor het tweede prototype van de Virtual LSD applicatie werd gezocht naar betere alternatieven voor het JMF raamwerk. Er werden verscheidene raamwerken getest die bovenop

⁵Quicktime, <http://www.apple.com/quicktime/>

Codevoorbeeld 4.7: Algoritme dat webcam-beelden tekent op het scherm met behulp van Processing en JMF.

```

45     private Player camPlayer;
46
47     private void loadCam() throws Exception{
48         CaptureDeviceInfo deviceInfo =
49             CaptureDeviceManager.getDevice("vfw:Microsoft WDM Image Capture (Win32):0");
50         camPlayer = Manager.createRealizedPlayer(deviceInfo.getLocator());
51         FormatControl formatControl =
52             (FormatControl) camPlayer.getControl("javax.media.control.FormatControl");
53         formatControl.setFormat ( deviceInfo.getFormats()[0]);
54         camPlayer.start();
55     }
56
57
58     public void draw() {
59         if(camPlayer != null){
60             FrameGrabbingControl fgc = (FrameGrabbingControl)
61                 camPlayer.getControl("javax.media.control.FrameGrabbingControl");
62             Buffer buf = fgc.grabFrame(); // Grabbing a frame
63             BufferToImage btoi =
64                 new BufferToImage((VideoFormat)buf.getFormat());
65             Image img = btoi.createImage(buf);//Creating Image from Buffer
66             if(img != null){
67                 PImage frame = new PImage(img);
68                 image(frame, 1, 1);
69             }
70         }

```

Processing draaiden. Een overzicht van deze raamwerken is te zien in de lijst van compatibele raamwerken op de website van Processing⁶. Uit deze raamwerken werd GSVideo gekozen voor zijn performantie en eenvoudig gebruik.

In codevoorbeeld 4.8 zien we de code die nodig is om de webcam-beelden op te vragen en te tekenen op het scherm. Op regel 9 wordt het *GSCapture*-object gedefinieerd dat in staat is voor het aanspreken van de webcam. De initialisatie van dit object staat beschreven op regel 12. In tegenstelling tot het JMF raamwerk, hoeft men de naam van de webcam niet mee te geven. De enige parameters die gevraagd worden om het object te initialiseren is de huidige klasse en de gewenste grootte van de webcam-beelden. Deze aanpak is duidelijk eenvoudiger dan met het JMF raamwerk. Op de regels 17-19 van codevoorbeeld 4.8 zien we hoe deze webcam-beelden opgevraagd (en getekend) worden. De conditie, dat beschreven staat op regel 17, gaat na als het *GSCapture*-object kan aangesproken worden. Door de *read*-methode op te roepen (regel 18), zullen alle oproepen naar het *GSCapture*-object hetzelfde webcam-beeld krijgen. Dit geeft meer controle aan de ontwikkelaar. Met de *get*-methode (regel 19) wordt het webcam-beeld opgevraagd als *PImage* (i.e. een datastructuur voor afbeeldingen in Processing). In de Virtual LSD applicatie werd dit *PImage*-object niet louter getekend, maar geanalyseerd via verscheidene technieken. De technieken die meer informatie halen uit het opgehaalde webcam-beeld worden besproken in de komende secties.

⁶Overzicht compatibele raamwerken voor Processing, <http://processing.org/reference/libraries/>

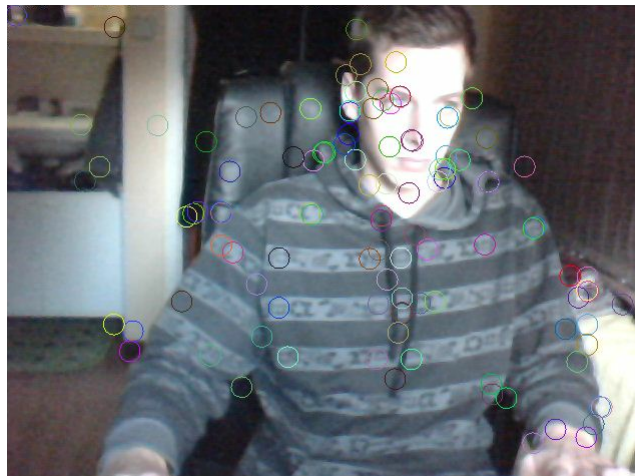
Codevoorbeeld 4.8: Algoritme dat webcam-beelden tekent op het scherm met behulp van Processing en GSVideo.

```
9     private GSCapture capture;  
10  
11    private void loadGScam(){  
12        capture = new GSCapture(this, 640, 480);  
13        capture.start();  
14    }  
15  
16    public void draw (){  
17        if(capture.available()){  
18            capture.read();  
19            image(capture.get(),0,0);  
20        }  
21    }  
22  
23    public void setup(){  
24        size(640, 480, P3D);  
25        loadGScam();  
26    }
```

4.3.2 *Interest points* detecteren

Het detecteren van *interest points* is al eerder besproken in 4.1.1. Tijdens de ontwikkeling van het eerste prototype werden de *interest points* gevisualiseerd door middel van kleurrijke objecten. Het resultaat hiervan is te zien in figuur 4.9. De gedetecteerde *interest points* blijken niet echt overeen te komen met hoeken en randen. Na verdere ontwikkeling van de Virtual LSD applicatie werd gekozen om contouren te gebruiken in plaats van *interest points*. We bespreken het detecteren van contouren in volgende sectie.

Figuur 4.9: Gedetecteerde *interest points* worden gevisualiseerd in verscheidene kleuren.



4.3.3 Contourdetectie

Het detecteren van contouren werd al toegelicht in 4.1.2. In de Virtual LSD applicatie werd het *Canny* detectie-algoritme geïmplementeerd. In codeVoorbeeld 4.10 is de imple-

mentatie van dit algoritme beschreven. Het BoofCV-raamwerk laat toe om de verschillende stappen van *Canny* detectie-algoritme, zoals eerder besproken, te gebruiken en te configureren naar eigen wens. De eerste stap van het algoritme bestaat uit een Gaussiaanse vervaging. Daarna worden de Gradiënten berekend met Sobel operatoren. De gepaste objecten worden hiervoor aangemaakt (regels 78-81). Het *CannyEdgeContourDynamic*-object voert het algoritme uit. De gedeclareerde Gaussiaanse filter en de gradiënt worden samen met twee drempelwaardes verwacht bij de initialisatie (regels 83-84) van dit object. De *getContours*-methode geeft de contouren in het beeld terug als een lijst van punten.

Codevoorbeeld 4.10: Detectie van contouren in een beeld met behulp van het *Canny* detectie-algoritme uit het BoofCV-raamwerk.

```

78 BlurFilter<ImageFloat32> blur =
79     FactoryBlurFilter.gaussian(ImageFloat32.class, 1, 10);
80 ImageGradient<ImageFloat32, ImageFloat32> gradient =
81     FactoryDerivative.sobel(ImageFloat32.class, ImageFloat32.class);
82
83 edgeDetector = new CannyEdgeContourDynamic<ImageFloat32, ImageFloat32>
84     (blur, gradient, (float)thresh*0.1f, (float)thresh);
85 edgeDetector.process(input);
86 contours = edgeDetector.getContours();

```

4.3.4 Bewegingsdetectie

Beweging in de webcam-beelden wordt eenvoudig geïmplementeerd door een verschil te nemen tussen twee opeenvolgende frames. Dit verschil wordt berekend door de pixelwaardes in het vorige en huidige frame te vergelijken. Als de kleurwaardes van een pixel in het huidige frame sterk verschillen van die in het vorig frame, wordt deze pixel gemarkeerd als "bewogen". Alle "bewogen" pixels krijgen hun originele kleur, terwijl de andere pixels zwart worden. Deze lijst van pixels (*Image*-datastructuur) wordt bijgehouden als de beweging in het frame. Als een visualisatie de beweging in een frame opvraagt, kan de visualisatie de gekleurde (niet-zwarte) pixels gebruiken. Als twee opeenvolgende frames hetzelfde beeld tonen, zullen alle pixels zwart blijven en is er dus geen beweging in het beeld aanwezig.

4.3.5 Huiddetectie

Een (goedkoop) alternatief voor het detecteren van gezichten of mensen is het herkennen van een huidskleur. Om de huid te detecteren van mensen die in het webcam-beeld voorkomen, werd een filter uit het Marvin-raamwerk⁷ gebruikt. De *SkinColorDetection*-filter haalt regio's uit het webcam-beeld die een kleur hebben dat sterk lijkt op de blanke huidskleur van mensen. Enkel de pixels met een gepaste kleur worden overgehouden in de filter. Net zoals bij de bewegingsdetectie, worden de andere pixels zwart gelaten. De filter uit het Marvin-raamwerk is niet heel nauwkeurig, maar dit is geen vereiste om een visualisatie te kunnen maken in de Virtual LSD applicatie. Een visualisatie die deze informatie gebruikt, is te zien in 5.3.3.

⁷Marvin Image Processin Framework, <http://marvinproject.sourceforge.net>

Hoofdstuk 5

Visualisatie

De twee vorige hoofdstukken beschreven onderzoeksdomeinen die informatie uit respectievelijk muziek en afbeeldingen halen. Deze informatie zal in dit hoofdstuk gebruikt worden om zinvolle visualisaties te tekenen op het scherm. We beginnen met een algemene situering van (informatie)visualisatie. Daarna halen we enkele raamwerken aan die ons toelaten om zinvolle visualisaties te tekenen. Tenslotte bespreken we de implementatie van het visualisatiegedeelte van de Virtual LSD applicatie. Hierin worden alle geïmplementeerde visualisaties besproken.

5.1 Situering

Het visualiseren van muziek kan gezien worden als een specifieke vorm van informatievevisualisatie. Muzikale data is een verzameling van verschillende soorten gegevens. Veel van deze gegevens worden beschreven over de tijd, zoals tempo, ritme, harmonie, etc. Het visualiseren van dergelijke muzikale data wordt meestal voorgesteld in 2D of 3D door deze tijdsafhankelijkheid. Gegevens die niet beschreven worden over de tijd, zoals metadata of de *beat sum*, stellen doorgaans statische (1D) informatie voor. Schneiderman[47] deelt informatievevisualisatie op in 7 datatypes. Degene die nuttig zijn voor de Virtual LSD applicatie zijn vooral *temporal*, *1D*, *2D* en *3D*. Door de eerder vermelde tijdsafhankelijkheid zijn de meeste audio features *temporal* data. Het verschil bij deze data is dat items een bepaalde start-en eindtijd hebben en dat items kunnen overlappen. Door het gebruik van een vensterfuncties (sectie 3.1.2) wordt deze data opgedeeld in intervallen waarbij telkens een 1D waarde bij is opgeslagen. Deze transformatie maakt *temporal* data bruikbaar als parameters. De webcam-beelden die opgehaald worden zijn 2D lijsten van pixels. Bij elke pixels worden kleurwaardes (doorgaans RGB) bijgehouden, die tezamen het kleur van een pixel representeren. Ook de output van de Virtual LSD applicatie is 2D. Elk frame dat naar de AR bril gestuurd wordt bevat opnieuw een dergelijke 2D lijst van pixels. In één visualisatie, *3DCloud*, werd het 3D datatype gebruikt. Van het webcam-beeld (2D) wordt een vlak gemaakt in een 3D ruimte. Punten in dit vlak zullen door middel van kleur en muzikale data verplaatst worden in deze 3D ruimte. De visualisatie wordt verder besproken

in sectie 5.3.3.

Volgens Kosara[24] valt muziekvisualisatie in het kamp van artistieke visualisatie in plaats van pragmatische visualisatie. Pragmatische visualisatie is een technische voorstelling om data te kunnen analyseren. Het doel van artistieke visualisatie is om een bepaald idee of gevoel over te brengen in plaats van data te tonen. De data wordt doorgaans getransformeerd in iets dat zichtbaar, interessant en verstaanbaar is. In de Virtual LSD applicatie zullen visualisaties een gevoel moeten overbrengen die past bij de muziek.

5.2 Raamwerken

5.2.1 Adobe Flash

Flash is een propriëtaire ontwikkelingsomgeving van Adobe voor het creëren van animaties en *multimediacontent*. Het was oorspronkelijk ontworpen door Jonathan Gay als FutureSplash Animator, waarmee animaties konden gemaakt worden door *keyframing* toe te passen op vectorafbeeldingen. Bij *keyframing* wordt tussen twee tekeningen of afbeeldingen een interpolatie uitgevoerd om tussenliggende frames te genereren. Hierdoor wordt de animator ontlast bij het maken van animatiefilms[6]. Naarmate de software evolueerde en meermaals van eigenaar veranderde, werd het steeds meer gebruikt voor bredere doeleinden. Door de introductie van de Actionscript programmeertaal, zijn de mogelijkheden van de ontwikkelingsomgeving sterk gestegen. Tegenwoordig wordt Adobe Flash vooral gebruikt om interactieve webapplicaties, prototypes, webvideo's en reclame te maken.

De ontwikkelingsomgeving biedt mogelijkheden aan artistiek-gerichte gebruikers en aan programmeurs. In de Flash (Professional) IDE kan een gebruiker animaties en eenvoudige webapplicaties maken, zonder kennis te hebben van programmeren. Vooral de tijdslijn waarop *keyframing* wordt toegepast, maakt dit mogelijk. Door de geïntegreerde programmeertaal is het ook mogelijk om met behulp van code, een applicatie te maken. Doordat Flash binnen een door Adobe ontwikkeld programma wordt geïnterpreteerd en uitgevoerd, is het bruikbaar op verschillende types van systemen zoals Mac, Windows en Linux. Het gratis programma waarmee Flash wordt uitgevoerd (i.e. *Flash Player*) is namelijk bij meer dan 96%¹ van de internet-toegankelijke *personal computers* aanwezig[17].

5.2.2 Processing

Processing is een *open source* programmeertaal en ontwikkelingsomgeving dat in 2001 gecreëerd werd als digitale schetsboek. Het doel van Processing was het aanleren van de beginselen van programmeren met een visuele context. Naarmate de ontwikkelingsomgeving en de *community* evolueerde, werd Processing voor steeds bredere doeleinden gebruikt. Tegenwoordig wordt Processing vooral gebruikt voor het ontwikkelen van sterk visuele prototypes.

¹Flash Player statistieken, <http://www.adobe.com/products/flashplatformruntimes/statistics.html>

De eenvoud van de programmeertaal zorgt voor een lage leercurve. In de Processing IDE worden verscheidene zaken verborgen gehouden voor de gebruiker. Dit laat minder ervaren programmeurs toe om met de software te kunnen werken. Naast de Processing IDE, is er ook een integratie van Processing met de Eclipse² IDE mogelijk. Hiermee kan doorgaans meer structuur geschapt worden om meer geavanceerde programma's te schrijven. Binnen Eclipse kan de Java programmeertaal gehanteerd worden, waar volledige integratie met Processing mogelijk is. Net zoals Java, is Processing *cross-platform*. Dit geeft aan dat de software kan gebruikt worden op verschillende types van systemen.

De enorme *community* achter Processing zorgt ervoor dat de ontwikkelingsomgeving evolueert, maar ook bijzonder toegankelijk is. Talloze *open source* raamwerken worden bovenop Processing ontwikkeld. Door de *open source*-cultuur worden veel projecten met elkaar gedeeld, waardoor codevoorbeelden talrijk aanwezig zijn[46].

5.2.3 OpenFrameworks

OpenFrameworks is een *open source* raamwerk dat voor C++ ontwikkeld werd om het creatieve proces bij experimenten te ondersteunen. Het raamwerk is gebaseerd op de principes van Processing, waardoor deze twee raamwerken sterk op elkaar gelijken. Ook dit raamwerk kan rekenen op een grote *community* die bijkomende functionaliteit schrijft en projecten deelt.

Volgens de ontwikkelaars van het raamwerk, is OpenFrameworks een uitbreiding op de Processing ontwikkelingsomgeving. Er is extra functionaliteit voorzien en er wordt meer controle gegeven aan de gebruikers. Zo kan polymorfisme (i.e. klassen laten overerven van andere, waar men extra functionaliteit aan kan geven) gebruikt worden in OpenFrameworks, dat bijvoorbeeld niet mogelijk is bij Processing. Net zoals Processing is dit raamwerk *cross-platform*, waardoor het bruikbaar is op verschillende types van systemen. In tegenstelling tot Processing wordt de code bij OpenFrameworks omgezet naar een vorm die bruikbaar is voor een ander systeem (e.g. Linux). Delen die moeilijk kunnen omgezet worden, zijn vooral in additionele raamwerken te vinden.

Het punt waar OpenFrameworks op hamert, is vooral performantie. Het raamwerk biedt een sterkere integratie met *OpenGL* aan in vergelijking met Processing. Deze integratie laat toe om visualisaties te tekenen met de GPU³. Ook interactie met *OpenCV* is sterk uitgebouwd. Hiermee onderscheidt OpenFrameworks zich van Processing, door een krachtigere en meer uitgebreide oplossing aan te bieden. Dit heeft echter een nadelig gevolg op de leercurve van het raamwerk[28].

5.2.4 Vergelijking

Voor het ontwikkelen van de Virtual LSD applicatie werd de Processing ontwikkelingsomgeving gekozen. De lage leercurve en de actieve *community* zorgen ervoor dat een prototype op korte tijd kan geïmplementeerd worden. Door de integratiemogelijkheden met

²Eclipse, <http://www.eclipse.org/>

³Graphics Processing Unit

Java, kunnen de andere verkozen raamwerken, BoofCV en jMIR, geïntegreerd worden met elkaar.

Adobe Flash werd niet gebruikt in deze masterproef om verscheidene redenen. Door de propriëtaire aard van de ontwikkelingsomgeving ligt de kost behoorlijk hoog. Bovendien wordt Flash beperkt in performantie doordat *Flash Player* in de browser wordt uitgevoerd. Een browser heeft namelijk minder rechten dan een programma dat uitgevoerd wordt op het besturingssysteem zelf. Tenslotte biedt Flash minder integratiemogelijkheden aan met andere raamwerken en programmeertalen dan het gekozen Processing.

OpenFrameworks lijkt een waardig alternatief voor Processing. Dit raamwerk zou vooral geschikt zijn om een verder uitgewerkte versie van de Virtual LSD applicatie te maken. De extra functionaliteit en hogere performantie zouden cruciaal kunnen zijn bij commerciële ontwikkeling voor de Virtual LSD applicatie. Het doel van deze masterproef was echter om een *proof-of-concept* applicatie te maken, die zijn nut en impact op muziekbeleving zou aantonen door evaluaties. Voor de vooropgestelde doelstelling leek het complexere Openframeworks niet nodig en werd gekozen voor Processing.

5.3 Implementatie

5.3.1 Tekenen met Processing

De Virtual LSD applicatie steunt op het Processing raamwerk om visualisaties te genereren. Om gebruik te kunnen maken van dit raamwerk, moet een klasse aangemaakt worden dat van *PApplet* overerft. Indien men hiervan overerft, is het verplicht om de methodes *setup* en *draw* te implementeren. De *PApplet* klasse kan beschouwd worden als een applicatie waar men kan tekenen op het scherm. Om de performantie hoog te houden, voorziet Processing een animatie-*thread* die instaat voor het tekenen. Alle zaken die moeten getekend worden, staan in de *draw* methode, dat elk frame wordt uitgevoerd. De *setup* methode wordt slechts éénmaal uitgevoerd, wanneer de applicatie gestart wordt. In deze methode worden de instellingen van de applicatie beschreven. In codevoorbeeld 5.1 zien we op regel 8 de beschrijving van deze instellingen. De instellingen die verplicht zijn om te beschrijven zijn de grootte (i.e. breedte en lengte) en de *renderer* van de applicatie. De *renderer* is de grafische *engine* die instaat voor het tekenen. Er is keuze voor de *JAVA2D*, *P2D*, *P3D* en *OPENGL* renderer. Voor de Virtual LSD applicatie werd *P3D* gekozen. Deze renderer beschikt over de functionaliteit om 3D-objecten te tekenen en geeft meer aandacht aan snelheid dan aan afwerking. Er werd niet gekozen voor de *OPENGL* renderer, die gebruik maakt van de GPU. Bij het vergelijken van de performantie tussen de *render engines* leek *P3D* de snelste te zijn. De verklaring waarom de *OPENGL* renderer niet de meest performante *renderer* is, ligt er in dat de code niet geoptimaliseerd is voor GPU-gebruik. Processing laat slechts beperkte interactie met *OPENGL* toe. Om volledig gebruik te maken van de GPU, zou men de ingebouwde functionaliteit van Processing kunnen overschrijven. De ontwikkelaar raadt dit echter ten zeerste af[46].

Het tekenen van vormen kan eenvoudig worden geprogrammeerd door de regels 12 en

```

1 package be.robbywauters;
2
3 import processing.core.*;
4
5 public class Virtual_LSD extends PApplet{
6
7     public void setup(){
8         size(640, 480, P3D);
9     }
10
11    public void draw (){
12        fill(random(255), random(255), random(255));
13        rect(random(width), random(height), 20, 20);
14    }
15
16    public static void main(String[] args) {
17        try {
18            PApplet.main(new String[] { "--exclusive", "be.robbywauters.Virtual_LSD" });
19        } catch (Exception e) {
20            e.printStackTrace();
21        }
22    }
23 }

```

Codevoorbeeld 5.1: Basisapplicatie met gebruik van Processing raamwerk

13 van codevoorbeeld 5.1. Hier staat *fill* voor het toewijzen van een kleur en *rect* voor het tekenen van een rechthoek. De basisapplicatie dat te zien is in codevoorbeeld 5.1, zal elk frame een gekleurd vierkant tekenen op een willekeurige plaats.

Doordat we Processing aanspreken in Eclipse in plaats van de Processing IDE⁴, moeten we onze klasse (hier *Virtual_LSD*) op een alternatieve manier opstarten. In de *Main* methode van de klasse moet de *PApplet* klasse aangesproken worden. In regel 18 van codevoorbeeld 5.1 zien we hoe de applicatie moet worden opgestart (vanuit de Eclipse IDE).

5.3.2 Basisstructuur van een visualisatie

In de Virtual LSD applicatie worden meerdere visualisaties beschreven. Alle visualisaties hebben een gemeenschappelijke basis en werken op dezelfde wijze. Hier zullen we de basisstructuur van een visualisatie bespreken, zodat we de basis niet steeds hoeven te herhalen bij de bespreking van de geïmplementeerde visualisaties.

Alle visualisaties worden beschreven in een Enumeratie-klasse, genaamd *VisualizationState*. Elke visualisatie heeft een tekenmethode dat elk frame wordt uitgevoerd. Binnen deze tekenmethode kan er real-time gebruik gemaakt worden van beschikbare audio features en de webcam-beelden. De structuur van een visualisatie bestaat uit de volgende onderdelen:

1. Huidig webcam-beeld opvragen
2. Extra omgevingsinfo opvragen

⁴Integrated Development Environment: software om programma's mee te beschrijven, coderen en organiseren

- (a) Contouren in het webcam-beeld
 - (b) Delaunay triangulatie van het webcam-beeld
 - (c) Beweging in het webcam-beeld
3. Audio features ophalen
 4. Pixelwaardes aanpassen op basis van audio features
 - (a) Kleuren kiezen
 - (b) Webcam-beeld tekenen
 - (c) Vormen tekenen
 - (d) Filterfunctie toepassen

De onderdelen 1 en 2 vragen data op aan de *CV Thread*. Alle berekeningen die te maken hebben met computer visie worden afgehandeld in deze *thread* (verder besproken in 4.3). Hiermee koppelen we de computer visie los van de visualisatie. Daarenboven maakt het een modulaire opbouw van de visualisaties mogelijk. De geïmplementeerde visualisaties zijn hierdoor makkelijk uitbreidbaar. Om een visualisatie te kunnen creëren die zich aanpast aan de omgeving van de gebruiker, kan het gebruik maken van 2a contouren, 2b Delaunay triangulatie of 2c beweging in het webcam-beeld.

Onderdeel 3 beschrijft het ophalen van audio features. De audio features dienen ook als parameter om een visualisatie aan te passen. In dit geval wordt de visualisatie aangepast aan de muziek van de gebruiker. De audio features kunnen opgehaald worden door de methode *getMusicFeature* aan te roepen, die beschreven staat in afbeelding 3.3.

Aan de hand van de audio features en omgevingsinfo kan een brede waaier van visualisaties gemaakt worden (onderdeel 4). De kleuren zullen voornamelijk bepaald worden door de audio features. Terwijl de vormen (e.g. driehoeken) sterk afhangen van de omgevingsinfo (e.g. Delaunay triangulatie). Verscheidene visualisaties maken gebruik van een filterfunctie die toegepast wordt op het beeld.

5.3.3 Geïmplementeerde visualisaties

In deze sectie zullen de geïmplementeerde visualisaties van de Virtual LSD applicatie overlopen en besproken worden. Het finaal prototype van de applicatie telt 7 uiteenlopende visualisaties. Zoals aangehaald in vorige sectie, hebben ze allemaal dezelfde basisstructuur. Er wordt een korte beschrijving gegeven van de verschillende visualisaties waarin verwijzingen zullen zitten naar de basisstructuur van vorige sectie. Naast de beschrijving, zal ook een motivatie gegeven worden die uitlegt waarom de visualisatie gekozen werd voor de Virtual LSD applicatie. Omdat een afbeelding soms meer zegt dan 1000 woorden, zal er van elke visualisatie een afbeelding (uit de Virtual LSD applicatie) opgenomen worden op het eind van deze sectie. De visualisaties van het eerste prototype zijn te zien in figuur 5.3, de visualisaties die werden toegevoegd in het tweede prototype zijn te zien in figuur 5.4.

Triangulatie

De eerste visualisatie die ontwikkeld werd in de Virtual LSD applicatie was de *triangulatie* (figuur 5.3a). Deze visualisatie maakt een benadering van het webcam-beeld door driehoeken te tekenen op het scherm. Deze driehoeken worden gecreëerd door een Delaunay triangulatie uit te voeren op de contouren van de omgeving. In een eerste implementatie werden lokale features van het webcam-beeld gebruikt om de triangulatie te berekenen. Maar contouren en hoeken zijn belangrijk voor het herkennen van objecten[3] en blijken betere resultaten te geven. De aantal driehoeken die gevisualiseerd worden op het scherm worden beperkt door (een weging van) de *Standaard Deviatie van de Spectral Centroid* audio feature. Hiermee worden er meer driehoeken getekend als de muziek helder lijkt. De kleur van de driehoeken wordt bepaald door de kleur te nemen van het middelpunt van de driehoek. De driehoeken hebben ook een variërende transparantie, dat gebaseerd is op de *Root Mean Square* audio feature. Hoe luider/krachtiger de muziek speelt, hoe meer de driehoeken zullen ingekleurd zijn. Deze driehoeken komen voor het webcam-beeld te liggen en benaderen het beeld met driehoeken. Bij een luide hoge toon wordt het webcam-beeld dus benaderd door een groter aantal kleinere driehoeken. Als we een luide bass hebben, wordt het beeld ruwer verdeeld in grotere driehoeken. Hiermee worden 2 muzikale percepties visueel duidelijk gemaakt door het aantal driehoeken en hun transparantie.

Fish-eye effect

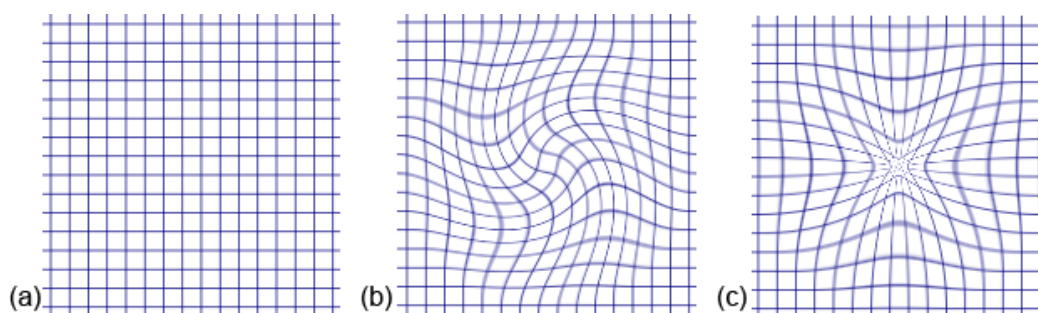
In een videoclip van Placebo (Meds⁵) die de effecten van LSD toont, kwamen enkele effecten voor die het beeld vervormen. Men kan dit zien als een *fish-eye* effect dat slechts op een deel van het beeld is toegepast (figuur 5.3b). Voor dit effect werd gekeken naar beeldfilters die beschikbaar waren in open source raamwerken. Een raamwerk die een uitgebreide lijst van (voor Java bestemde) beeldfilters aanbied is JH Labs⁶. Voor het *fish-eye* effect werd de *Pinch* filter gebruikt uit dit raamwerk. De *Pinch* filter laat toe om een *Pinch* en *Whirl* effect te creëren. In figuur 5.2 zien we hoe een Whirl-effect het beeld draait rond een punt (b). Bij het Pinch-effect wordt het beeld naar het punt toe getrokken (c). Deze visualisatie maakt geen gebruik van omgevingsinfo zoals contouren of beweging.

3D Cloud effect

Het derde en laatste effect dat aanwezig was in het eerste prototype van de Virtual LSD applicatie, is het *3D Cloud* effect (figuur 5.3c). In deze visualisatie wordt het webcam-beeld opgesplitst in cellen. Het aantal cellen hangt af van de grootte van de cellen. De grootte van de cellen wordt berekend op basis van de *Zero Crossings* audio feature. Deze feature geeft een indicatie van de hoeveelheid ruis en verandert bij sterk verschillende tonen (indicatie van pitch). De cellen kunnen beschouwd worden als pixelregio's van bijvoorbeeld 4 op 4 pixels groot. Deze cellen worden in een 3D ruimte geplaatst op een vlak dat parallel

⁵Videoclip: Placebo - Meds, <http://www.youtube.com/watch?v=pxQ10IDsLJI>

⁶JH Labs, <http://www.jhlab.com/>



Figuur 5.2: Overzicht van het Fish-eye effect. Het originele beeld (a), het beeld waar een Whirl-effect(b) en een Pinch-effect(c) is op uitgevoerd

staat met het beeld. Als we in deze fase van de visualisatie zouden tonen, zien we een ruw gepixeld beeld dat iets verder lijkt te staan dan het webcam-beeld. Deze cellen (of pixelregio's) worden op basis van hun kleur en de *Root Mean Square* audio feature verplaatst in de z-richting. Dit heeft het effect dat de cellen naar de gebruiker toe komen. Omdat deze visualisatie gebruik maakt van de kleur van de cellen om de translatie te bepalen, wordt het totale beeld vervormd. Als we de kleuren niet in rekening zouden brengen, komt eenvoudigweg het gepixelde webcam-beeld naar de gebruiker toe. Deze visualisatie geeft een zeer sterke indruk weer van de kracht dat in de muziek zit. Een liedje met een luide en frequente bass komt hiermee helemaal tot zijn recht.

Bewegingsdetectie

Het visualiseren van de beweging (figuur 5.4a) is waarschijnlijk de eenvoudigste visualisatie van de Virtual LSD applicatie. In 4.3.4 werd uitgelegd hoe de beweging uit de webcam-beelden kon geëxtraheerd worden. Deze beweging wordt hier gevisualiseerd door alle pixels te kleuren die bewogen bleken te zijn. De kleur van deze pixels werd bepaald door de audio features *Beat Sum*, *Standaard deviatie van de Spectral Centroid* en *Standaard deviatie van de Spectral Smoothness*. Om overgangen tussen kleuren vloeiend te maken, werden de kleurwaardes gekozen in de HSB⁷ kleurruimte. De *brightness*-waarde werd ingevuld door de perceptuele helderheid van de muziek (*Standaard deviatie van de Spectral Centroid*). Voor de kleurwaardes werd de *hue*-waarde aangepast door een combinatie van de *Beat Sum* en *Standaard deviatie van de Spectral Smoothness* audio features. De *saturation*-waarde bleef constant op, zodat steeds kleurrijke waarden zouden verkregen worden.

Dizziness effect

Het *dizziness* effect (figuur 5.4b) kan los vertaald worden als het effect van de dronken man. In deze visualisatie wordt het webcam-beeld vertraagd getoond en lijken meerdere opeenvolgende webcam-beelden getoond te worden. Als je bijvoorbeeld met de hand voor de webcam zal wuiven, zal deze visualisatie het pad van de hand tonen op het beeld. De

⁷Hue, Saturation en Brightness

kleuren in het webcam-beeld worden ook licht aangepast op basis van de *Zero Crossings* en *Spectral Centroid* audio feature.

Dit effect geeft zeer goeie feedback gekregen tijdens de evaluaties, maar is verrassend eenvoudig te implementeren. Elk webcam-beeld krijgt een licht aangepaste kleur door de *tint* methode van Processing. In deze *tint* methode kunnen we ook een *alpha*-waarde (transparantie) meegeven. Als we een *alpha*-waarde meegeven die kleiner is dan 1, worden opeenvolgende frames boven elkaar getekend. Maar doordat ze een zekere transparantie hebben, blijft het vorig getekende frame nog steeds deels zichtbaar. Hoe kleiner we deze *alpha*-waarde kiezen, hoe trager ons beeld lijkt te zijn en hoe meer vorig getekende frames steeds zichtbaar zijn.

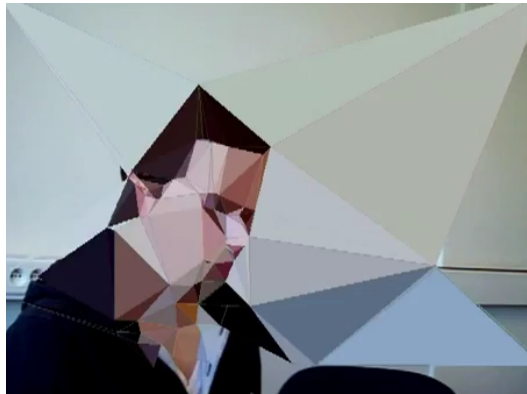
Huiddetectie

In 4.3.5 werd uitgelegd hoe de huidskleur uit de webcam-beelden kon geëxtraheerd worden. De regio's die als huid gecategoriseerd werden (verder vernoemd als huidregio's), worden in deze visualisatie in de verf gezet (figuur 5.4c). Op basis van de *Spectral Flux* en *Root Mean Square* audio features wordt het webcam-beeld en de huidregio's met elkaar gemengd. Deze menging is geïmplementeerd als verscheidene *blending modes*. Bij het gebruiken van een *blending mode* wordt een beeld (e.g. het webcam-beeld) aangepast op basis van een ander beeld (e.g. huidregio's) dat er bovenop komt te liggen. Het webcam-beeld wordt hier aangepast door de kleurwaardes van elke pixel lineair te combineren met de pixels van de huidregio's.

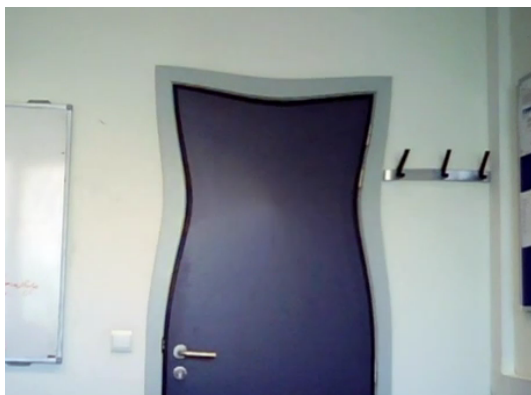
Processing geeft de mogelijkheid om meerdere *blending modes* te berekenen tussen twee beelden. In de Virtual LSD applicatie werden er hiervan twee toegepast. De eerste *blending mode* heet *Screen*. Hier zullen de inverse kleurwaardes van beide beelden vermenigvuldigd worden, waardoor de pixels resulteren in een lichtere versie. *Screening* met zwart laat de kleur van het andere beeld onveranderd, terwijl *screening* met wit een witte kleur produceert. De andere heet *Hard Light*. Deze *blending mode* vertrekt van een grijze kleur die even ver van wit als van zwart verwijderd ligt. Als de kleur van de pixels in het bovenste beeld lichter is dan deze grijze kleur, wordt *screening* toegepast. In het andere geval wordt de omgekeerde bewerking uitgevoerd, namelijk het vermenigvuldigen van de kleurwaardes van beide beelden. De huiddetectie geeft een zwarte afbeelding weer waar enkel de huidregio's een kleur krijgen. Hierdoor worden alle pixels van webcam-beeld donkerder gemaakt die niet gecategoriseerd werden als huid. Bij *Hard Light* wordt dus vooral de huid getoond. Door een afwisseling van deze *blending modes* ontstaat een wat lugubere visualisatie.

Contrasterende contouren

De laatste visualisatie (figuur 5.4d) bestaat uit twee delen. In het eerste deel wordt het contrast van het webcam-beeld aangepast op basis van de *Root Mean Square* audio feature. Het aanpassen van het contrast wordt geïmplementeerd als een filter die afkomstig is van het Marvin raamwerk. Als de muziek luider wordt, zal het contrast in het webcam-beeld



(a) Triangulatie

(b) *Fish-eye effect*(c) *3D Cloud*

Figuur 5.3: De visualisaties uit het eerste prototype van de Virtual LSD applicatie.

ook toenemen. Een toename van contrast zorgt ervoor dat kleuren feller worden en het verschil tussen donker en licht vergroot. Het andere deel tekent de contouren van het vorig webcam-beeld. We nemen het vorig webcam-beeld omdat we hierdoor een effect verkrijgen dat de notie van tijd lijkt te verstoren. Dit effect is gelijkaardig aan het *dizziness* effect, maar minder hevig. Doordat de contouren slechts een frame achterkomen, blijft de visualisatie responsief. De kleur van de contouren wordt hier aangepast op basis van de *Standaard Deviatie van de Spectral Smoothness*.



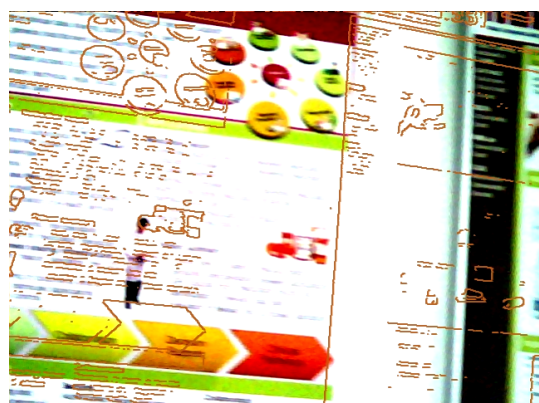
(a) Visualisatie van bewegingsdetectie



(b) *Dizziness* effect



(c) Visualisatie van huiddetectie



(d) *Contrasterende contouren* effect

Figuur 5.4: De visualisaties uit het tweede prototype van de Virtual LSD applicatie.

Hoofdstuk 6

Evaluatie

De vorige hoofdstukken beschreven de architectuur en de drie onderzoeksdomeinen hierbinnen, waarmee meerdere prototypes werden gemaakt. In dit hoofdstuk zullen we de evaluaties bespreken van deze prototypes. De nadruk binnen de evaluaties ligt op immersie en beleving. Om dit te kunnen meten wordt eerst een nuttige methode gezocht en ontwikkeld. Met deze ontwikkelde methode, worden twee evaluatiesessies uitvoerig besproken.

6.1 Eigen definitie van immersie

In sectie 2.5.5 werd een eigen definitie gegeven aan immersie. Deze definitie was gebaseerd op gelijkaardige concepten en bestaande definities. We herhalen hier de vijf kerncomponenten die immersie definiëren:

- *Attention*: De mate waarin de concentratie en de aandacht van de gebruiker naar de applicatie gaat. Irrelevante informatie zoals alledaagse bekommernissen worden doorgaans weggefilterd.
- *Enjoyment*: Plezier beleven met de applicatie. Alsook de mate waarin esthetiek (e.g. gebruik van gepaste vormen en kleuren) bijdraagt tot de beleving.
- *Emotional Attachment*: Gevoelens (e.g. blijdschap of angst) die vrijkomen door het gebruiken van de applicatie.
- *Transportation*: Het verlies van zelfbewustzijn en het geloof dat de virtuele wereld realistisch is.
- *Temporal dissociation*: De tijd die ervaren wordt in de virtuele omgeving lijkt niet dezelfde als die van de reële omgeving (Verstoorde notie van tijd).

Deze vijf kerncomponenten zullen in dit hoofdstuk verder gebruikt worden om immersie en beleving te evalueren bij gebruikers van de Virtual LSD applicatie.

6.2 Methodes voor het evalueren van beleving en immersie

Voor het evalueren van systemen of applicaties zijn er enkele methodes beschikbaar[39]. Zo hebben we:

- Questionnaires
- Gebruikerstesten in een lab
- Evaluatie door experts
- Meten van gebruikersgegevens

Questionnaire

Gebruikers een vragenlijst laten invullen na het gebruik van een functioneel prototype is een veel gebruikte methode. Doordat mijn evaluatie specifiek gericht is op beleving en immersie tijdens het gebruik van mijn applicatie, zijn bestaande questionnaires, e.g. QUIS¹, USE² en SUS³ minder van toepassing.

In het domein van Virtual Reality zijn er verscheidene questionnaires beschikbaar die wetenschappelijk onderbouwd zijn. Een veel gebruikte is de Presence questionnaire [60](Zahorik Jenison, 1998), [50](Slater - Usoh, 1993, 1994), [58](Witmer - Singer, 1998). Deze questionnaire heeft enkele tekortkomingen waardoor ze minder geschikt is voor de vooropgestelde doelstelling; 1) er heerst onenigheid over de focus en het doel[49](Slater, 1999. Measuring presence), 2) de questionnaire meet 'presence', wat slechts een onderdeel is om immersie te beschrijven, 3) het onderzoek is specifiek gericht op Virtual Environment, waarbij de gebruiker enkel een virtuele omgeving ziet, terwijl Augmented Reality een combinatie is van de echte en virtuele omgeving. Omdat het evalueren van immersie in Augmented Reality een relatief jong onderzoeksgebied is, werd een eigen questionnaire opgesteld. In de volgende sectie vindt u verdere informatie over het opstellen van de questionnaire.

Gebruikerstesten in een lab

Slater[50] benadrukt dat immersie vereist dat er een relatie moet zijn tussen de proprioceptieve feedback (i.e. beweging van de spieren) van de gebruiker en de informatie die gegenereerd is op de displays. De gebruiker filmen tijdens het gebruiken van de applicatie zou dus waardevolle data kunnen opleveren. Als we dit combineren met het registreren van wat er gelijktijdig op de displays te zien is, hebben we een krachtige manier om verbanden te leggen tussen de bewegingen van de gebruiker als reactie op de gegenereerde

¹Questionnaire for User Interface Satisfaction

²Usefulness, Satisfaction and Ease of use

³System Usability Scale

beelden. Deze manier werd toegepast tijdens de eerste evaluatiesessie van mijn applicatie. Het meten van proprioceptieve feedback werd echter niet gebruikt in de evaluatiesessies.

Evaluatie door experts

Door een beperkt budget is het moeilijk om externe experts in te huren voor een onderzaging. Doordat dit een relatief jong onderzoeksdomein is, is het sowieso een moeilijke taak om een expert in het gebied te hebben. Mijn persoonlijke kennis reikt niet ver genoeg om mezelf te beschouwen als expert in het evalueren van beleving en immersie in AR. Deze optie werd bijgevolg niet weerhouden tijdens mijn evaluatie.

Meten van gebruikersgegevens

Het meten van gebruikersgegevens moet hier gezien worden als het registreren van proprioceptieve en zintuiglijke feedback (e.g. bloeddruk, handpalmzweet, hartslag). Door het beperkte budget, leek het mij zeer moeilijk om apparatuur te gebruiken die deze feedback kon registreren. Bovendien wordt de gebruiker gefilmd tijdens het testen van de Virtual LSD applicatie. Hierdoor is het wel mogelijk om lichamelijke feedback waar te nemen.

In de tweede evaluatiesessie van de Virtual LSD applicatie, werd er getracht om elektroencefalogrammen (EEG) te meten van de gebruikers. Een analyse op de EEG zou moeten aantonen als er emotionele reacties aanwezig waren door het gebruik van de applicatie. Om deze EEG te meten, werd de commercieel beschikbare Emotiv Epoc gebruikt. Meer info over deze EEG headset vindt u in 2.3.5. De bevindingen van de tweede evaluatiesessie worden beschreven in sectie 6.5.

6.3 Het ontwikkelen van de questionnaire

Er is geen consensus over een questionnaire die moet gehanteerd worden om immersie te meten. Veel questionnaires zijn specifiek gemaakt voor een bepaald onderzoeksdomein (e.g. games) en er is veel discussie over het doel en de definitie van immersie[60][49][58]. In deze masterproef werd een eigen questionnaire ontwikkeld die de immersie van een gebruiker zou meten. Aan de basis hiervan liggen de vijf kerncomponenten die in sectie 6.1 werden herhaald. Deze componenten werden gedistilleerd uit concepten die gelijkaardig zijn aan immersie, zoals *Flow*, *CA* en *Presence*. Bij elk kerncomponent werden meerdere vragen opgesteld. Hieronder volgt een overzicht van de kerncomponenten waar we de vragen in opnemen. De nummers refereren naar de vragen in bijlage B. Hiermee worden de eigenschappen van deze questionnaire verder besproken.

- Attention: 1, 2, 3, 4
- Enjoyment: 5, 6
- Emotional Attachment: 7, 8, 9, 10, 11

- Transportation: 12, 13, 14, 15, 16, 17
- Temporal dissociation: 18, 19

Een deel van deze vragen zijn gebaseerd op een bestaande questionnaires rond immersie bij games[22] en Virtual Environments[58]. De vragen moeten ingevuld worden aan de hand van een Likert schaal[21]. Deze schaal van 1 tot 5 geeft de mate aan waarin een ondervraagde akkoord gaat met een gegeven stelling. Ter controle, werden zowel positieve als negatieve vragen opgenomen in de questionnaire. Bij een positieve vraag (e.g. “Tijdens het gebruik van de applicatie dacht ik aan niks anders dan het bewegen en controleren van de virtuele wereld.”) wordt een hoge score op de Likert schaal als goed bevonden. Als deze vraag negatief gesteld wordt (e.g. “Ik dacht aan andere zaken (alledaagse zaken) tijdens het gebruik van de applicatie.”), wordt een lage score als goed bevonden. Door de afwisseling van deze vragen doorheen een questionnaire kunnen de antwoorden van de ondervraagde gecontroleerd worden op consistentie. Sommige vragen die naar eenzelfde onderwerp polsen, werden op licht verschillende wijze gesteld. Hiermee is de robuustheid van een kerncomponent groter doordat er meerdere vragen gesteld worden[9]. De volledige vragenlijst is te zien in bijlage B.

Naast de 19 vragen omtrent immersie en beleving, werden nog 6 demografische vragen gesteld die iets meer zeggen over de ondervraagden. In deze vragen werden de leeftijd, het geslacht en de ervaring met computers, games en MP3-spelers opgenomen. Doordat er meerder prototypes van de Virtual LSD applicatie werde ontwikkeld en geëvalueerd, werd ook gevraagd als een ondervraagde al een voorgaande versie had getest. Met deze vragen kan een beeld geschept worden van een ondervraagde. Bij onverwachte resultaten kan deze extra informatie gebruikt worden om een conclusie te trekken.

6.4 Eerste evaluatiesessie

Bij de evaluatie van het eerste prototype werd vooral nagegaan als de applicatie een nut had en toegevoegde waarde kon betekenen bij de gebruikers. Er werd bekeken in welke mate de gebruikers beïnvloed werden in de aangepaste realiteit op basis van de kernbegrippen waarmee de questionnaire is opgebouwd (zoals besproken in vorige sectie).

6.4.1 Evaluatieverloop

De evaluatie van het eerste prototype werd uitgevoerd met 9 testgebruikers, waarvan 8 mannen en 1 vrouw. Alle mannelijke testgebruikers waren student, assistent of professor met een sterke informatica-achtergrond. De gemiddelde leeftijd van de testgebruikers lag tussen de 25 en 35 jaar. Uit het testpubliek waren er 3 personen die alreeds een voorbarige demoversie van de applicatie hebben gebruikt. Alle evaluaties werden op eenzelfde dag afgenomen. Er werden sessies van 25 à 45 minuten voorzien zodanig dat elke testgebruiker individueel kon getest en ondervraagd worden zonder invloed van andere gebruikers.

Een evaluatiesessie bestond uit 3 delen; eerst werd de applicatie getest door de gebruiker. Daarna werd er een questionnaire ingevuld. De sessie werd afgesloten met een interview.

Het eerste prototype van Virtual LSD bestond uit 3 opeenvolgende liedjes met elk een andere visualisatie. Het eerste liedje was Warriors Dance van The Prodigy, waarbij een triangulatie werd toegepast op de omgeving van de gebruiker. Daarna was Meds van Placebo te horen, waar er *fish-eye*-effecten gecreëerd werden op het beeld. Er werd ook gebruik gemaakt van een *glowfilter* die het beeld overbelichtte indien de muziek luider werd. Het laatste liedje was I need your lovin' van Jennifer Lowpass. In de laatste visualisatie werd het camerabeeld - die de gebruiker ziet - opgedeeld in cellen. Deze cellen werden op basis van de muziek en de kleur verplaatst in een 3D ruimte. Deze visualisaties kunt u zien in figuur 5.3 in het vorige hoofdstuk. Tijdens de testfase werd zowel de applicatie als de gebruiker zelf gefilmd. Door middel van Fraps⁴ konden de beelden die de applicatie genereerde opgenomen worden. Een videocamera filmde de gebruiker van op een kleine afstand tijdens de test. Hierdoor was het mogelijk om een link te leggen tussen bepaalde lichaamsreacties en gegenereerde beelden in de applicatie.

Nadat de gebruikers de applicatie getest hadden, werd er gevraagd om een questionnaire in te vullen. Zoals besproken in 6.3 bestaat deze uit 6 vragen die iets meer over de testgebruiker zeggen en 19 vragen over de applicatie die aan de hand van een Likert schaal moeten ingevuld worden[21]. Ter controle, werden zowel positieve als negatieve vragen opgenomen in de questionnaire. Sommige vragen die naar eenzelfde onderwerp polsen, werden op licht verschillende wijze gesteld. De volledige vragenlijst vindt u in bijlage B. Op het einde van de questionnaire waren er nog 3 open vragen, maar deze werden doorgeschoven naar het interview om een uitgebreider antwoord te krijgen van de gebruikers. De evaluatiesessie werd afgesloten met een interview. Hierin werd ondervraagd wat er bijdroeg aan de beleving en wat er storend was tijdens de test. Naargelang het gesprek werden er subvragen gesteld om een uitgebreider antwoord los te krijgen van de testgebruiker. Alle interviews werden opgenomen om daarna te kunnen verwerken. De gemiddelde duur van een interview was 14 minuten.

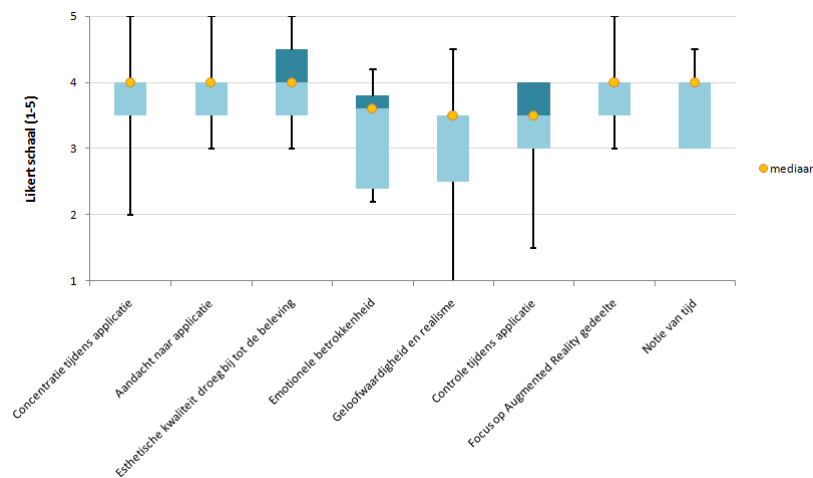
6.4.2 Resultaten

Om een beter beeld te krijgen op de resultaten uit de bevraging, zijn deze voorgesteld in een boxplotgrafiek. In bijlage C zien we een boxplotgrafiek die elke vraag van de questionnaire opneemt. Op de verticale as zien we de score op een Likert schaal (1 tot 5), op de horizontale as zijn alle vragen opgesomd. Er werd gebruik gemaakt van zowel positieve als negatieve vragen, i.e. vragen over hetzelfde onderwerp waar enerzijds een lagere score beter is en anderzijds een hogere score beter is. Hierdoor lijken de resultaten in deze boxplotgrafiek sterk te schommelen.

Om de resultaten beter te kunnen analyseren, zijn de vragen over eenzelfde onderwerp/-

⁴<http://www.fraps.com/>

Figuur 6.1: Samengevoegde Boxplotgrafiek



kernbegrip samengevoegd. Hierin werd rekening gehouden met de afwisseling van positieve en negatieve vragen, zodat voor alle resultaten een hogere score beter was. Zoals besproken in sectie 6.3, werden de vragen van de questionnaire ontwikkeld per kerncomponent. Om opnieuw een overzicht te verkrijgen per kerncomponent worden deze vragen opnieuw samengenomen.

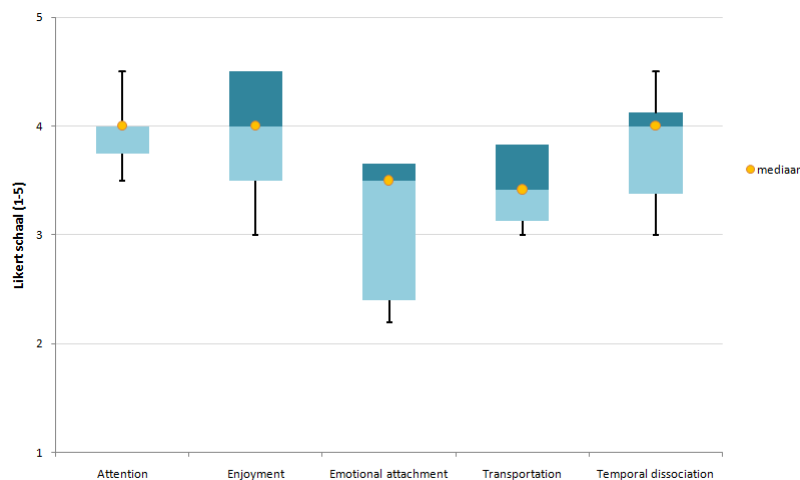
In figuur 6.1 is er opnieuw een boxplotgrafiek te zien, maar deze keer zijn er op de horizontale as kernbegrippen te zien in plaats van vragen. Deze kernbegrippen stellen een reeks van vragen voor die omgezet werden zodat een positieve score beter is. De kernbegrippen kunnen herleid worden naar de 5 onderdelen waarop dit onderzoek naar beleving en immersie in AR gebaseerd is. De kernbegrippen *concentratie* en *aandacht* vallen onder de noemer **Attention**. *Esthetische kwaliteit* hoort bij **Enjoyment**. Zo is er ook *Emotionele betrokkenheid* die bij **Emotional attachment** hoort. Het belangrijkste onderdeel in het onderzoek naar immersie in AR is **Transportation**. hierin zitten de begrippen *geloofwaardigheid en realisme*, *controle tijdens de applicatie* en *focus op het AR gedeelte* vervat. Als laatste kernbegrip hebben we *notie van tijd* die het onderdeel **Temporal dissociation** voor zijn rekening neemt.

In figuur 6.2 krijgen we een overzicht van de resultaten per kerncomponent. We zien op het eerste zicht een relatief hoge score voor elk kerncomponent. De mediaan ligt gemiddeld op 3,825. Voor een score van 1 tot 5 ligt dit voor een eerste prototype zeker niet slecht.

In het interview werd er gepolst naar storende factoren, elementen die bijdroegen tot de beleving en algemene voorkeuren en opmerkingen. De storende factoren die uit de interviews met de gebruikers kwamen worden hier opgesomd. Er werden meerdere keren gelijkaardige opmerkingen gegeven.

- De te lage framerate brak de beleving.

Figuur 6.2: Boxplotgrafiek per kerncomponent



- Visualisaties die te weinig op muziek reageerde.
- De echte wereld was te veel zichtbaar, doordat de bril niet nauw genoeg aansloot. Hierdoor kon er licht van buitenaf invallen op de ogen van de gebruiker.
- Grootte van 'het scherm' was te klein door het ontbreken van fullscreen-functionaliteit.
- Beperkt gezichtsveld dat zichtbaar was door de bril.

De resolutie van de bril was echter geen storende factor bij de testgebruikers. De bril waarmee getest werd had een resolutie van 800 op 600 pixels en de webcam waarmee beelden van de omgeving werden doorgestuurd had een resolutie van 640 op 480 pixels. De testgebruikers deelden allemaal dezelfde mening dat er geen pixels konden gezien worden of dat het tenminste niet storend was voor de beleving en immersie.

Er werd gepolst naar elementen die bijdroegen tot de beleving en geloofwaardigheid van de applicatie. Opnieuw werden hier meerdere gelijklopende opmerkingen gegeven.

- Responsiviteit: een vlotte framerate, zodat de bewegingen van de gebruiker en de visualisaties nauw aansluiten bij de realiteit.
- Een directe, duidelijke link tussen de visualisaties en de muziek die speelde.
- Soort visualisatie past meer bij een bepaald genre of specifiek liedje. Een goede keuze van de visualisatie passend bij de muziek die speelt, laat de visualisatie beter uitkomen.
- Het gebruik van Augmented Reality. Omdat men nog een deel van de echte wereld ziet, waren de gebruikers nog steeds capabel om mensen op te merken en in een ruimte te navigeren. Sommige testpersonen gaven aan dat een volledige afsluiting van de echte wereld, de applicatie minder bruikbaar zou maken.

- Het gebruik van een AR bril. Door de bril en de oortjes ben je opgesloten in een aparte wereld. Hierdoor kan je ten volle je aandacht besteden aan de muziek.

Als er gevraagd werd of de applicatie een meerwaarde bood aan het beluisteren van muziek of als de muziek intenser leek door een bijpassende visualisatie, waren alle gebruikers het erover eens. Er is wel degelijk een meerwaarde en de muziek lijkt intenser, op de strenge voorwaarde dat er een duidelijke link is tussen de visualisaties en de muziek. Een testgebruiker merkte wel op dat aan de meerwaarde ook een negatieve nuance moest toegevoegd worden. Omdat de visualisaties in dit prototype gebruik maakten van één muziek-feature, leken andere karakteristieken minder aandacht te krijgen. Hierdoor werd er voor een manipulatieve beleving gezorgd bij de gebruiker.

De voorkeursvisualisatie ging voor zes van de acht testgebruikers uit naar visualisatie 3, het *3D Cloud* effect. Ook hier was de duidelijke link tussen visualisatie en muziek het meest voorkomend argument. Ook de framerate van visualisatie 3 ging gemiddeld sneller dan de andere 2. Er werd aan visualisatie 1 en 2 elk één voorkeurstem toegekend.

Cruciaal in het interview was de vraag als de gebruiker ooit een soortgelijke applicatie zou gebruiken. Alle testpersonen zeiden ja, maar de meeste hadden wel enkele voorwaarden. Er waren enkele bedenkingen bij de context waarin je de applicatie zou gebruiken (achter het stuur, op de bus, thuis). Sommigen benadrukten de storende factoren die aan bod kwamen in deze evaluatie. Als de storende factoren zouden verholpen worden, zou dit de applicatie en de beleving sterk verbeteren. Enkele andere gebruikers waren erover eens dat ze de applicatie niet voor een lange duur zouden gebruiken.

6.4.3 Conclusie

In figuur 6.1 zien we dat de twee kernbegrippen die het laagst scoren *emotionele betrokkenheid* en *geloofwaardigheid en realisme* zijn. De lagere score van het laatste is vooral te wijten aan de storende factoren die opgesomd werden. De lage framerate, het ontbreken van een fullscreen functionaliteit en een beperkt gezichtsveld waren de grootste boosdoeners. Deze factoren zorgen dat de beleving gelimiteerd of gebroken wordt voor de gebruiker. De hardware van de prototype-opstelling heeft hier iets mee te maken. Maar de framerate zou vooral softwarematig moeten opgelost worden.

Het andere kernbegrip met een lage score, *emotionele betrokkenheid* heeft in zekere zin minder last van de opgesomde storende factoren. De applicatie lijkt dus minder invloed te hebben op de emotionele betrokkenheid van de gebruiker. Doordat het onderzoek steunt op meerdere begrippen, is dit geen totale ramp. Een verbetering in dit gebied zal moeilijk te behalen zijn, tenzij het oplossen van de storende factoren ook hierop een positieve invloed hebben. Een andere oorzaak voor deze lage score kan zijn dat gebruikers weigerachtig zijn tegen vragen over emoties. Tijdens de ondervraging bleken gebruikers dit onderwerp te mijden.

De antwoorden die uit het interview voort kwamen, waren vrij gelijklopend. Bijna

elke gebruiker vond de lage framerate en het ontbreken van de fullscreen-functionaliteit een storende factor. Het eerste probleem kan opgelost worden op verscheidene manieren. Zo kan er gebruik gemaakt worden van krachtigere hardware, efficiëntere algoritmes of kan ervoor gezorgd worden dat de GPU⁵ in staat is voor het tekenen. De fullscreen-functionaliteit kan eenvoudig opgelost worden. Deze functionaliteit was uitgeschakeld tijdens de evaluatie om mee te kunnen volgen op een externe monitor.

Bij de elementen die volgens de gebruikers bijdroegen tot de beleving en geloofwaardigheid zien we de vernoemde storende factoren. Als deze factoren zouden opgelost worden, zou dit een sterke bijdrage betekenen. Er werd ook vermeld dat het gebruik van AR wel degelijk voor een meerwaarde zorgt. Hier zien we dus een positieve impact van AR binnen dit onderzoek.

De testgebruikers waren erover eens dat er meerwaarde geboden werd bij de muziekbeleving door de applicatie. De muziek zou intenser overkomen en de aandacht zou meer naar de muziek getrokken worden. We zien ook hier de cruciale factor terugkomen dat er een duidelijke link moet zijn tussen visualisatie en muziek.

Op het einde van de evaluatie waren alle gebruikers ervan overtuigd dat ze een soortgelijke applicatie in de nabije toekomst zouden gebruiken. Dit toont aan dat de applicatie nuttig is en een toegevoegde waarde biedt aan de muziekbeleving. In een volgende iteratie zouden de storende factoren verder weggewerkt kunnen worden en zou de beleving nog groter worden.

6.5 Tweede evaluatiesessie

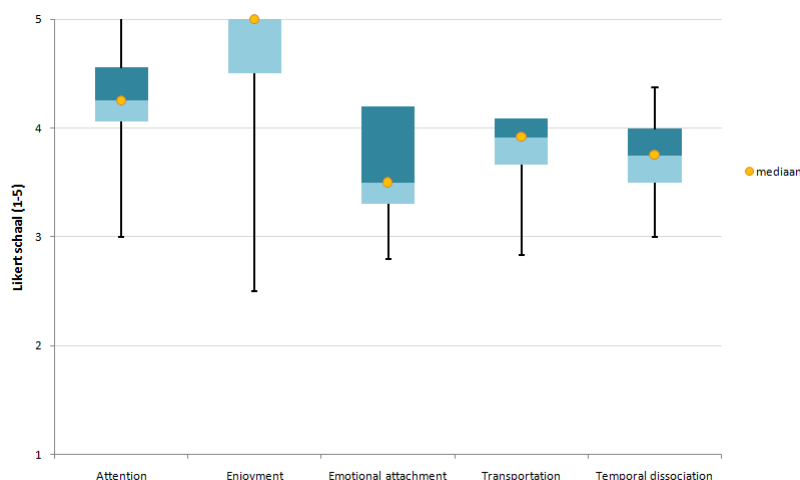
Bij de evaluatie van het tweede prototype werd vooral nagegaan als de de cruciale problemen uit de eerste evaluatie verholpen waren. Vooral de lage framerate en de soms onduidelijke link tussen visualisatie en muziek bleken de grootste problemen te zijn uit de vorige evaluatie. De Virtual LSD applicatie werd aangepast om deze zaken te verhelpen. Uit vorige evaluatie bleek dat het bepalen van de emotionele betrokkenheid in een ondervraging moeilijk is. In deze evaluatie werd getracht om met de Emotiv Epoc (EEG headset), emoties te registreren.

6.5.1 Evaluatieverloop

De evaluatie van het twee prototype werd uitgevoerd met 8 mannelijke testgebruikers. Alle testgebruikers waren student, assistent of professor met een sterke informatica-achtergrond. De gemiddelde leeftijd van de testgebruikers lag tussen de 25 en 35 jaar. Uit het testpubliek waren er vier personen die de vorige evaluatie hebben meegedaan en vier personen die nog nooit de applicatie getest hadden. Gelijkaardig aan de eerste evaluatie bestond een evaluatiesessie uit drie delen; eerst werd de applicatie getest door de gebruiker. Daarna werd er een questionnaire ingevuld. De sessie werd afgesloten met een interview. Bij vijf gebruikers werd tijdens het testen van de applicatie, de Emotiv Epoc gebruikt.

⁵Graphical Processor Unit

Figuur 6.3: Boxplotgrafiek per kernbegrip



Het tweede prototype van Virtual LSD bestond uit 10 opeenvolgende liedjes waar willekeurige visualisaties werden getoond. Er werd elke 20 seconden overgegaan naar 1 van de 7 verschillende visualisaties die geïmplementeerd waren. Naast de 3 visualisaties uit het eerste prototype, zijn nog vier visualisaties toegevoegd; *bewegingsdetectie*, *dizziness*, *huiddetectie* en *contrasterende contouren*. Deze visualisaties werden in vorig hoofdstuk uitgebreid besproken in sectie 5.3.3 en zijn te zien in figuur 5.4. Uit analyse van de eerste evaluatie bleek dat er uit het filmen van de gebruiker en de gegenereerde beelden weinig conclusies konden getrokken worden. In deze evaluatie werd dan ook beslist om dit niet meer toe te passen.

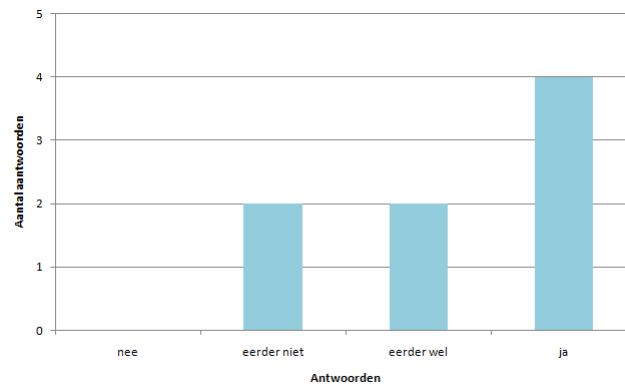
6.5.2 Resultaten

Om een beter beeld te krijgen op de resultaten uit de bevraging, zijn deze opnieuw voorgesteld in een boxplotgrafiek. In bijlage D zien we een boxplotgrafiek waarin opnieuw elke vraag van de questionnaire werd opgenomen. Op de verticale as zien we de score op een Likert schaal (1 tot 5), op de horizontale as zijn alle vragen opgesomd.

Ook deze boxplotgrafiek werd herleid naar een compactere weergave. In figuur 6.3 is een boxplotgrafiek te zien per kerncomponent (die we gebruiken om immersie te definiëren). In deze grafiek krijgen we een duidelijker beeld van de questionnaire. De mediaan (4,08) ligt gemiddeld hoger dan in de vorige evaluatie (3,825). Als we de vergelijking maken met figuur 6.2, zien we dat de scores voor (bijna) elk kerncomponent hoger zijn. De uitzondering hierop is *temporal dissociation*, dat lager lijkt te scoren. In volgende sectie worden de conclusies besproken die we hieruit kunnen trekken.

In het interview werd er opnieuw gepolst naar storende factoren, elementen die bijdroe-

Figuur 6.4: Antwoorden op de interviewvraag *Klonk de muziek intenser tijdens het gebruik van de Virtual LSD applicatie?*



gen tot de beleving en algemene voorkeuren en opmerkingen. Storende factoren waren hoofdzakelijk opmerkingen over de hardware. Geen enkele gebruiker leek de (onduidelijke) link tussen muziek en visualisatie storend te vinden. De framerate was voor de meesten voldoende, maar mocht nog steeds hoger liggen. Drie testgebruikers klaagden over het haperen van het geluid in de applicatie. Anderen hadden echter geen klachten hierover. De problemen in verband met de hardware, waren gelijklopend met de vorige evaluatie.

Er werd gepolst naar elementen die bijdroegen tot de beleving en geloofwaardigheid van de applicatie. Alle gebruikers waren positief over de verscheidenheid aan visualisaties en hoe deze op de muziek reageerden. Ook het continu afwisselen van de visualisaties viel zeer goed in de smaak.

Alle gebruikers vonden dat de applicatie een duidelijke meerwaarde gaf aan de muziek. De antwoorden op de vraag 'Klonk de muziek intenser?' zijn te zien in figuur 6.4. Hier zien we dat de meeste gebruikers de muziek als intenser beleefden.

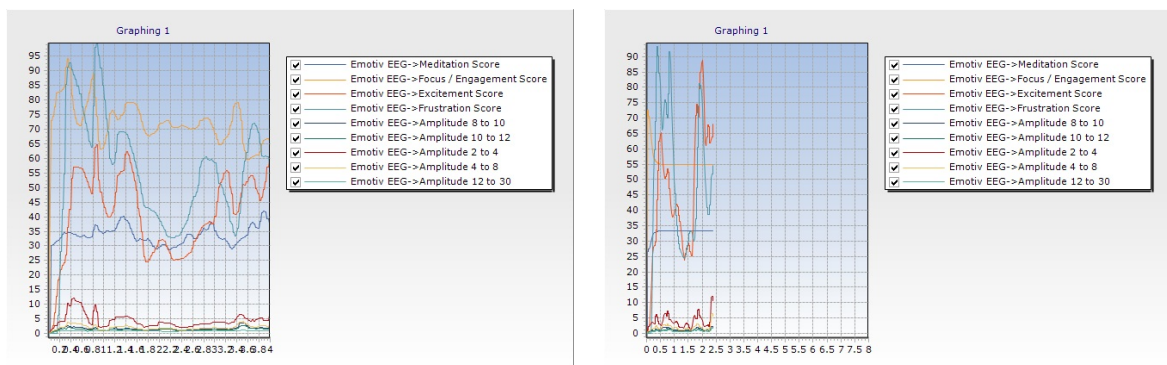
Het leek moeilijker in deze evaluatie om een voorkeursvisualisatie te kiezen. De gebruikers konden vooral de afwisseling appreciëren. Elke visualisatie werd wel door minstens één gebruiker vernoemt tijdens deze vraag. Dit toont aan dat de verschillende visualisaties aan elkaar gewaagd zijn qua kwaliteit (of amusement). Alle gebruikers waren enthousiast over de afwisseling aan visualisaties. Deze bevindingen geven aan dat de vernieuwingen in het tweede prototype geslaagd zijn.

Opnieuw werd er gevraagd of de gebruiker ooit een soortgelijke applicatie zou gebruiken. Tijdens deze ondervraging waren de testgebruikers meer uitgesproken. Allen zouden ze de applicatie nog willen gebruiken in de toekomst. Hierbij gingen ze er wel van uit dat de technologie van de AR bril zou verbeteren. Er werd door sommigen gesuggereerd waar ze de applicatie zouden gebruiken. Zowel thuis, onderweg met de fiets of bus, of op een concert werden vernoemd als mogelijke plaatsen.

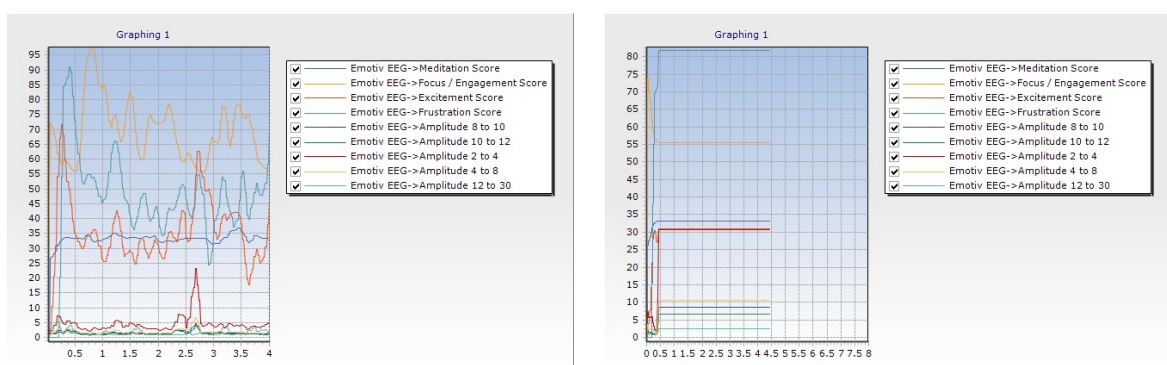
In deze evaluatiesessie werd getracht om emoties van de testgebruikers te meten met be-

hulp van een EEG headset. Bij vijf gebruikers werd tijdens het testen van de applicatie de Emotiv Epoc gebruikt om hersengolffrequenties op te meten. De combinatie van de AR bril en de EEG headset is mogelijk gebleken, maar maakte het oncomfortabel voor de testgebruiker. Met het softwarepakket Mind Workstation⁶ kunnen frequentiebanden uit een EEG headset gehaald worden. De Emotiv Epoc, die gebruikt werd in deze evaluatiesessie, is onder andere compatibel met deze software.

Uit de 5 sessies waar gebruik gemaakt werd van de Emotiv Epoc, waren er slechts 2 metingen bruikbaar voor analyse. Doordat de gebruiker (doorgaans) beweegt bij het gebruik van de Virtual LSD applicatie, zijn er regelmatig elektroden die hun connectie verliezen. Hierdoor worden de metingen onbetrouwbaar en laat de software Mind WorkStation enkel constante waardes zien. In figuur 6.6 zien we in de meest rechtse grafiek de constante waardes als horizontale lijnen. Dit maakt het zeer moeilijk om betrouwbare data uit deze evaluatiesessies te halen.



Figuur 6.5: Calibratiemetingen van testgebruikers met Mind WorkStation software en de Emotiv Epoc headset.



Figuur 6.6: Metingen tijdens de Virtual LSD applicatie met Mind WorkStation software en de Emotiv Epoc headset.

⁶Mind WorkStation van Transparent Corporation, <http://www.transparentcorp.com/>

Als calibratie werden metingen uitgevoerd op de testgebruikers voordat ze de Virtual LSD applicatie testten. De testgebruikers luisterden gedurende de meetperiode naar dezelfde muzieknnummers die gebruikt werden in de Virtual LSD applicatie. Hiermee kunnen we de metingen tijdens het gebruik van de applicatie (te zien in figuur 6.6) vergelijken met de calibratiemetingen (te zien in figuur 6.5).

Door de instabiele draadloze connectie en een beperkt aantal metingen per persoon, werd er onvoldoende data opgehaald om een statistisch onderscheid te maken met de baseline (de calibratiemetingen). Deze resultaten zullen dus niet gebruikt worden in de conclusie van deze sectie en het besluit van de masterproef. De weinige proefopnames laten wel de uitvoerbaarheid van een dergelijk experiment zien. Mits een zorgvuldige opstelling en voldoende proefpersonen en sessies, kan de hypothese 'EEG frequentiebanden verschillen tijdens een immersie-experiment van een baseline' getoetst worden.

6.5.3 Conclusie

Als we figuur 6.2 en figuur 6.3 naast elkaar leggen, zien we in de tweede evaluatie een duidelijke verbetering bij zowat alle kerncomponenten. De enige uitzondering hierop is *temporal dissociation*. Dit komt doordat de semantiek van de vragen in de questionnaire, het concept (van verstoorde notie van tijd) wat tegenspreekt. Bij het opstellen van de questionnaire werd er enkel van uit gegaan dat de notie van tijd als sneller zou ervaren worden. Dit is te zien in de verklaring "*Het voelde voor mij aan dat een zeer korte tijd is voorbij gegaan*". Maar in het tweede prototype zijn er visualisaties (e.g. *Dizziness* en *Contrasterende contouren*) die de tijd perceptueel trager doen verlopen.

De hoge scores op de kerncomponenten tonen aan dat de verbeteringen die doorgevoerd werden, geslaagd zijn. De snellere framerate werd verkregen door efficiëntere algoritmes en raamwerken. De link tussen visualisatie en muziek leek duidelijk door het gebruiken van meerdere audio features in de visualisaties.

Hoofdstuk 7

Besluit

In dit laatste hoofdstuk worden de vooropgestelde doelen, die besproken werden in 1.4, getoetst aan de behaalde resultaten. Bovendien wordt ook een kritische reflectie gegeven over de aanpak van deze masterproef. Tenslotte worden enkele pistes aangehaald die kunnen leiden tot verder onderzoek.

7.1 Doelstellingen

In deze masterproef werd een *proof-of-concept* AR applicatie gemaakt die muziek visualiseert in de omgeving van de gebruiker. Elk vooropgesteld doel wordt hier vergeleken met de bereikte resultaten.

Onderzoeken van de architectuur

In sectie 2.2 werd de architectuur van de Virtual LSD applicatie kort toegelicht. Deze architectuur is ook opgenomen in bijlage A. Hier zijn alle belangrijke componenten aangeduid met bijbehorende raamwerken die in deze masterproef gebruikt werden. Uit onderzoek is gebleken dat meerdere domeinen nodig waren om de vooropgestelde applicatie te kunnen ontwikkelen. De bevindingen uit de uitgevoerde evaluatiesessies tonen aan dat de combinatie van *Music Information Retrieval (MIR)*, *Computer Visie (CV)* en *Visualisatie* nodig zijn om een dergelijke applicatie te maken. De architectuur hangt samen met componenten en raamwerken uit deze domeinen. Voor MIR zijn er twee raamwerken nodig, één voor het ophalen van informatie uit muziek en één voor het afspelen van die muziek. Bij het CV-gedeelte werden vier verschillende raamwerken gebruikt; één voor het ophalen van de webcam-beelden, één voor het detecteren van contouren en *interest points*, één voor het berekenen van triangulaties en tenslotte één raamwerk dat instaat voor huiddetectie. Voor de Visualisatie waren drie raamwerken nodig; de belangrijkste was deze die het mogelijk maakt om iets te tekenen op het scherm, daarnaast waren er nog twee raamwerken die filters aanboden. Uit een onderzoek met aanzienlijk veel raamwerken werden deze negen verkozen en aan elkaar gekoppeld.

Onderzoeken van een hardware-opstelling

Om een systeem te ontwikkelen dat gebruik maakt audio, video en AR, is een set van bepaalde apparaten nodig. In sectie 2.3 werd een opstelling gegeven die een AR bril, webcam en laptop aan elkaar koppelde. Zoals besproken in deze sectie waren verscheidene alternatieven mogelijk. De opstelling van de gekozen apparaten bleek in staat te zijn om een audiovisueel AR systeem functioneel te maken.

Ontwikkelen van een *Proof-of-concept* AR applicatie

Aan de hand van de vooropgestelde architectuur (bijlage A) is een prototype ontwikkeld dat de muziek kan visualiseren in de omgeving. Er werd vooral gestreefd naar het bouwen van een *Proof-of-concept* applicatie om na te gaan of het een toegevoegde waarde kan betekenen. Bij de implementatie ging minder aandacht naar het robuust en uitbreidbaar maken van de software. De ontwikkeling van de prototypes stond in het teken van de evaluaties die ermee zouden worden uitgevoerd.

In het eerste prototype werden drie verschillende visualisaties ontwikkeld en getest; *triangulatie*, *fish-eye* en *3D Cloud*. Deze visualisaties gebruikten informatie uit de omgeving en de muziek en waren opgesteld aan de hand van vaste structuur. Deze structuur die de basis vormde voor alle visualisaties, werd besproken in sectie 5.3.2. Na het ontwikkelen van dit prototype werd een eerste evaluatiesessie gehouden. Deze duidde vooral de essentiële componenten aan die ontbraken. De componenten die naar voor kwamen, waren 1) een duidelijke link tussen visualisatie en muziek, 2) een vlotte framerate, 3) de mogelijkheid om te navigeren door de echte wereld en 4) de link tussen de visualisatie en opgewekte emotie/sfeer.

Bij het tweede prototype werden deze componenten in het achterhoofd gehouden bij het doorvoeren van verbeteringen en uitbreidingen. Hier waren in totaal zeven visualisaties aanwezig die nog meer informatie uit de muziek en omgeving gebruikten. Naast de bestaande drie visualisaties, werden nog vier nieuwe ontwikkeld; *bewegingsdetectie*, *dizziness*, *huiddetectie* en *contrasterende contouren*. Bij de evaluatiesessie met dit tweede prototype werd geoordeeld dat de componenten die ontbraken in het eerste prototype, hier wel aanwezig waren. De duidelijke link tussen visualisatie en muziek werd beter geïntegreerd door meerdere audio features te gebruiken bij de visualisaties. De framerate werd dan weer verhoogd door efficiëntere algoritmes en raamwerken te gebruiken, alhoewel hier nog ruimte is voor verbetering.

Onderzoeken van methodes om immersie te meten

In sectie 2.5 werden verscheidene concepten besproken die sterk gelijken op immersie. Met de bevindingen van dit onderzoek werd een eigen definitie gegeven aan immersie die gebruikt werd om een questionnaire op te stellen. Deze definitie van immersie steunde op 5 kerncomponenten; *attention*, *enjoyment*, *emotional attachment*, *transportation* en *temporal dissociation*. De ontwikkelde questionnaire werd in beide evaluatiesessies gebruikt en duidde aan waar er verbeteringen mogelijk waren in de applicatie. Naast een questionnaire

werden ook andere methodes getest die een maatstaf kunnen zijn voor immersie. Zo werd de gebruiker gefilmd tijdens de eerste evaluatiesessie en werd een vergelijking uitgevoerd met de gegenereerde beelden van applicatie. Deze aanpak leverde echter weinig bruikbare informatie op. Tijdens de tweede evaluatiesessie werd bij vijf testpersonen de Emotiv Epoc gebruikt om aan de hand van hersengolffrequenties iets meer te kunnen zeggen over hun emotionele reacties. Dit werd echter niet verdergezet bij de andere testpersonen nadat enkel twee van deze vijf evaluaties bruikbare data opleverden. Het meten van deze frequenties vergt meer onderzoek en voorbereiding dan mogelijk was binnen deze masterproef. Uit de gebruikte methodes bleek de eigen questionnaire uiteindelijk de meest nuttige om immersie te meten.

Integratie van de omgeving onderzoeken aan de hand van immersie

Pas in de tweede evaluatiesessie was het mogelijk om de mobiliteit van de gebruikers te testen. In tegenstelling tot de eerste sessie, konden de testpersonen hier vrij rondlopen bij het beleven van de Virtual LSD applicatie. Uit de afgenomen tests is gebleken dat gebruikers capabel zijn om op een voorzichtige manier te navigeren door hun omgeving en hierin andere mensen op te merken. De resultaten van de questionnaires gaven aan dat de gebruikers wel degelijk een vorm van immersie ervoeren tijdens het gebruik van de applicatie. Verscheidene testpersonen gaven tijdens de interviews aan dat de combinatie van een echte omgeving met virtuele objecten, de beleving ten goede kwam en een duidelijke meerwaarde bood.

Aantonen van het nut van de applicatie

Door de ontwikkeling van een degelijk prototype, was het haalbaar om het nut van de Virtual LSD applicatie aan te tonen. Er werd vooral informatie omtrent het nut van de applicatie gehaald uit de interviews tijdens de twee evaluatiesessies. In de eerste evaluatiesessie zagen alle testgebruikers het zitten om de applicatie later te gebruiken, maar daar waren echter veel voorwaarden en bedenkingen aan verbonden. Bij de tweede evaluatiesessie hadden de testgebruikers een meer uitgesproken mening en gaven ze duidelijk aan dat ze een dergelijke applicatie (zeer) graag zouden gebruiken. Sommigen gingen hier verder op in en duiden mogelijke plaatsen aan, waar ze de Virtual LSD applicatie zouden willen gebruiken. Zowel thuis als onderweg met de fiets of de bus werden genoemd als mogelijke plaatsen waar de applicatie een toegevoegde waarde kon zijn. Als we deze bevindingen vergelijken met het vooropgesteld doel om muziekvisualisatiesoftware naar mobiele computergebruikers te brengen, komen we al sterk in de buurt om deze doelgroep te bereiken. De ondervraagden gaven ook aan dat de applicatie een meerwaarde bood aan het beluisteren van muziek en hierbij werd vaak vermeld dat de muziek ook intenser klonk.

Aantonen van de praktische haalbaarheid van de applicatie

Ondanks de primitieve en *low-budget* opstelling van de Virtual LSD applicatie, bleek de hardware toch in staat een (quasi-)immersieve audiovisuele ervaring te creëren. Er kwa-

men tijdens de evaluatiesessies enkele klachten over de hardware naar boven, maar geen van deze is onoverkomelijk. Aan de hardware werd tussen het eerste en tweede prototype niks veranderd, maar recente ontwikkelingen van commerciële bedrijven[48] geven aan dat deze problemen snel zouden opgelost zijn. De commercieel toegankelijke AR brillen zijn nog steeds niet op het niveau gekomen dat een applicatie zoals Virtual LSD zonder klachten gebruikt kan worden. De interesse van het grote publiek en enkele grote spelers zoals Google[40] en Sony[32], brengen hier hoogst waarschijnlijk in de nabije toekomst verandering in.

7.2 Kritische reflectie

Uit de evaluatiesessies van de prototypes kwamen heel wat klachten naar boven omtrent de gebruikte hardware. Vooral de AR bril leek enkele beperkingen te hebben die storend waren voor de gebruikers. Voorbeelden hiervan zijn het beperkte gezichtsveld. Ondanks de primitieve hardware-opstelling waren de testgebruikers wel enthousiast over de Virtual LSD applicatie. Het concept werd ook tijdens presentaties, op de blog en tijdens evaluaties goed ontvangen. De recente ontwikkelingen in AR hardware en de interesse van grote spelers zoals Google[40] geven aan dat een dergelijk idee een toegevoegde waarde kan bieden in de nabije toekomst.

Het meten van immersie blijft een punt waar minder uitsluitel kan over gegeven worden. De ontwikkelde questionnaire heeft een waardevolle dienst bewezen bij het evalueren en verbeteren van de applicatie. Desalniettemin is er onvoldoende bewijs geleverd dat de questionnaire een (vol)waardige methode is voor het meten van immersie en moeten andere methodes overwogen worden.

De gebruikte raamwerken bleken voldoende functionaliteit aan te bieden die nodig was om een *proof-of-concept* applicatie te ontwikkelen. Alternatieve raamwerken die meer functionaliteit en performantie voorzien, zijn sterk te overwegen bij het bouwen van een meer uitgebreide applicatie. De framerate die cruciaal bleek te zijn voor de beleving, werd niet altijd als voldoende bevonden. Dit was een moeilijkheid binnen deze masterproef waar van het begin tot het einde aan gewerkt werd. De (soms) gebrekkige framerate is een gevolg van de geïmplementeerde algoritmes, gekozen raamwerken en gebruikte hardware. Hier is zeker en vast een verbetering nodig als een dergelijke applicatie verder zou ontwikkeld worden.

De muziek en bijpassende visualisaties synchroon laten lopen, bleek een harde noot om te kraken. Er werden hiervoor meerdere raamwerken en algoritmes getest, zoals vermeld in secties 3.3.2 en 3.3.3. Door middel van een apart hulpprogramma werd een correct werkend algoritme ontwikkeld die in staat was om real-time audio features te plotten op het scherm.

In sectie 3.1.4 werd de relatie tussen muziek, kleur en emotie onderzocht. Alhoewel verscheidene concepten (e.g. vrolijk klinkende muziek wekt dezelfde emotie op als felle gele en groene kleuren) aanneembaar zijn, bleek het zeer moeilijk om dit in de visualisaties te verwerken. Door de geringe integratie van deze concepten, kunnen hierover weinig conclusies genomen worden. Binnen het domein van *muziek, kleur en emotie* zijn noch-

tans interessante onderzoeken mogelijk die gerelateerd zijn aan deze masterproef. Zo zou bijvoorbeeld de impact van kleur op de muziekbeleving en opgewekte emotie kunnen bestudeerd worden. Alsook de manipulatie van muziekbeleving door visualisaties specifieke emoties te laten overbrengen.

7.3 Mogelijke toekomstige uitbreidingen

In de toekomst zou een onderzoek kunnen gaan naar het sociale aspect van dergelijke applicaties die in groep worden gebruikt. Het concept van de omgeving aanpassen op basis van muziek, kan overgebracht worden naar bijvoorbeeld een festival waar elke aanwezige een soortgelijk systeem bezit. De visualisaties hierbij zouden dan door de organisator of optredende muziekgroep zelf bepaald kunnen worden.

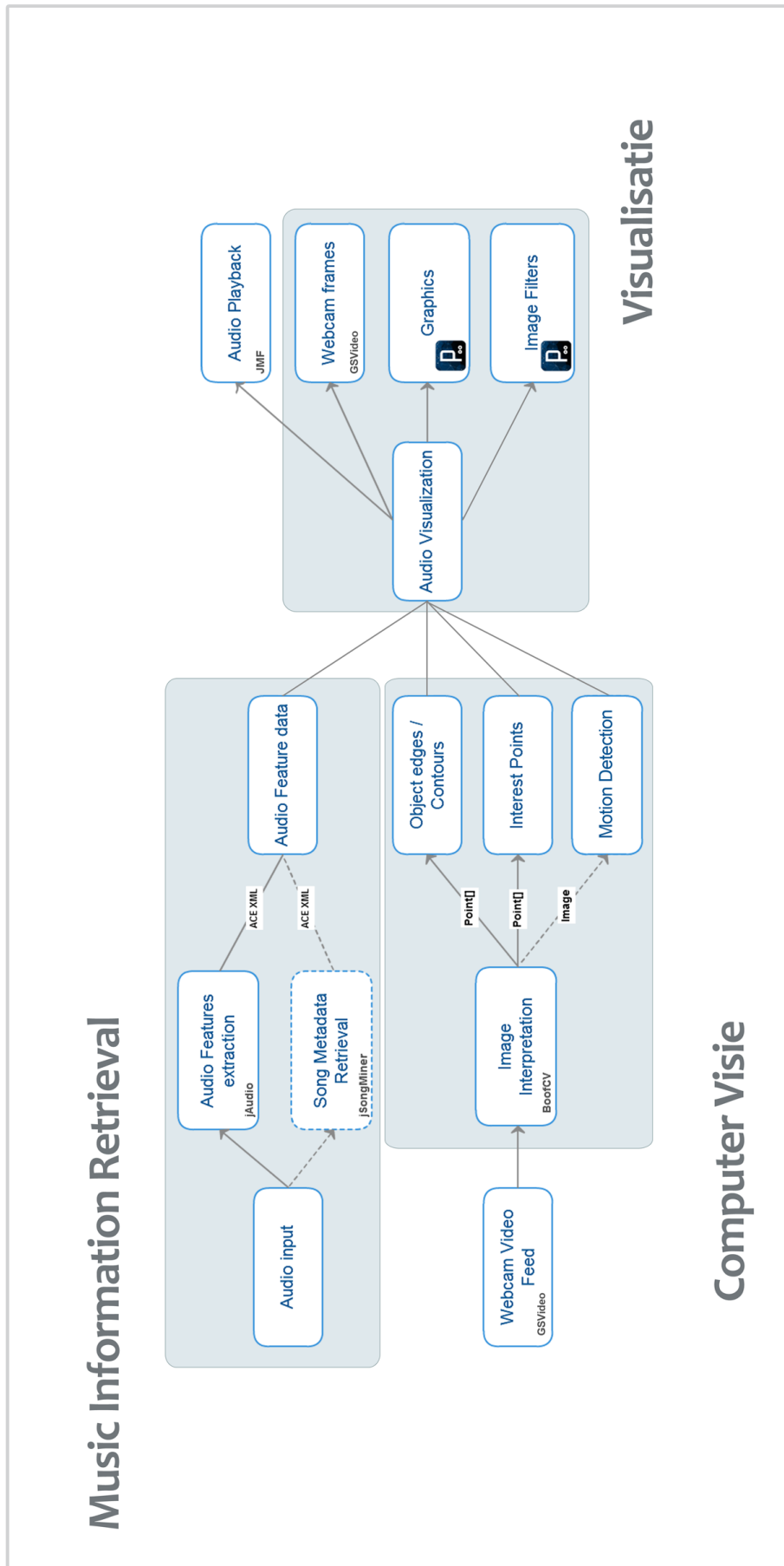
Een uitbreiding op de bestaande applicatie zou kunnen gebruik maken van objectherkenning en deze informatie gebruiken om relevante visualisaties te genereren. Een voorbijrijdende auto zou bijvoorbeeld een andere visualisatie genereren dat een persoon die voorbij wandelt. Voor een robuuste werking van een dergelijke *computer visie*-integratie moeten objectcategorieën kunnen herkend worden. Hiermee is een systeem in staat om een wijde reeks van objecten te herkennen in plaats van enkel specifieke objecten.

Een onderdeel van deze applicatie dat naar boven kwam in de brainstormsessies, is een interface die de muziek kan besturen. Bijvoorbeeld door het gebruik van *hand gestures* kan bijvoorbeeld naar een volgend liedje worden gegaan, zodat directe interactie met een smartphone of MP3-speler overbodig wordt. Deze functionaliteit werd echter niet verder uitgebouwd. De ontwikkeling van een dergelijk systeem zou een toekomstige uitbreiding kunnen zijn.

Bijlage A

Architectuur van de Virtual LSD applicatie

De architectuur van de Virtual LSD applicatie, die de drie grote onderzoeksdomeinen schetst; *Music Information Retrieval, Computer Visie en Visualisatie*



Bijlage B

Questionnaire

1. Ik moest me zeer sterk concentreren tijdens het gebruiken van de applicatie.
2. Tijdens het gebruik van de applicatie dacht ik aan niks anders dan het bewegen en controleren van de virtuele wereld.
3. Ik had moeite om mij te concentreren tijdens het gebruiken van de applicatie.
4. Ik dacht aan andere zaken (alledaagse zaken) tijdens het gebruik van de applicatie.
5. De esthetische kwaliteit van de applicatie droeg bij tot de beleving.
6. De esthetische kwaliteit van de applicatie was een storende factor.
7. De applicatie gaf me een gelukkig gevoel.
8. De applicatie gaf me een onrustig gevoel.
9. Ik voelde een emotionele reactie tijdens het gebruiken van de applicatie.
10. Ik voelde geen emotionele betrokkenheid tijdens het gebruiken van de applicatie.
11. De applicatie deed me niks.
12. Ik waande de beleving als echt.
13. De beleving was op geen enkel moment geloofwaardig.
14. Ik had het gevoel dat ik de graphics controleerde.
15. De controle tijdens de applicatie had ik volledig niet in handen.
16. Ik hield me meer bezig met de gegenereerde graphics dan de echte wereld.
17. Ik zag vooral de echte omgeving tijdens het gebruik van de applicatie.
18. Het voelde voor mij aan dat een zeer korte tijd is voorbij gegaan.

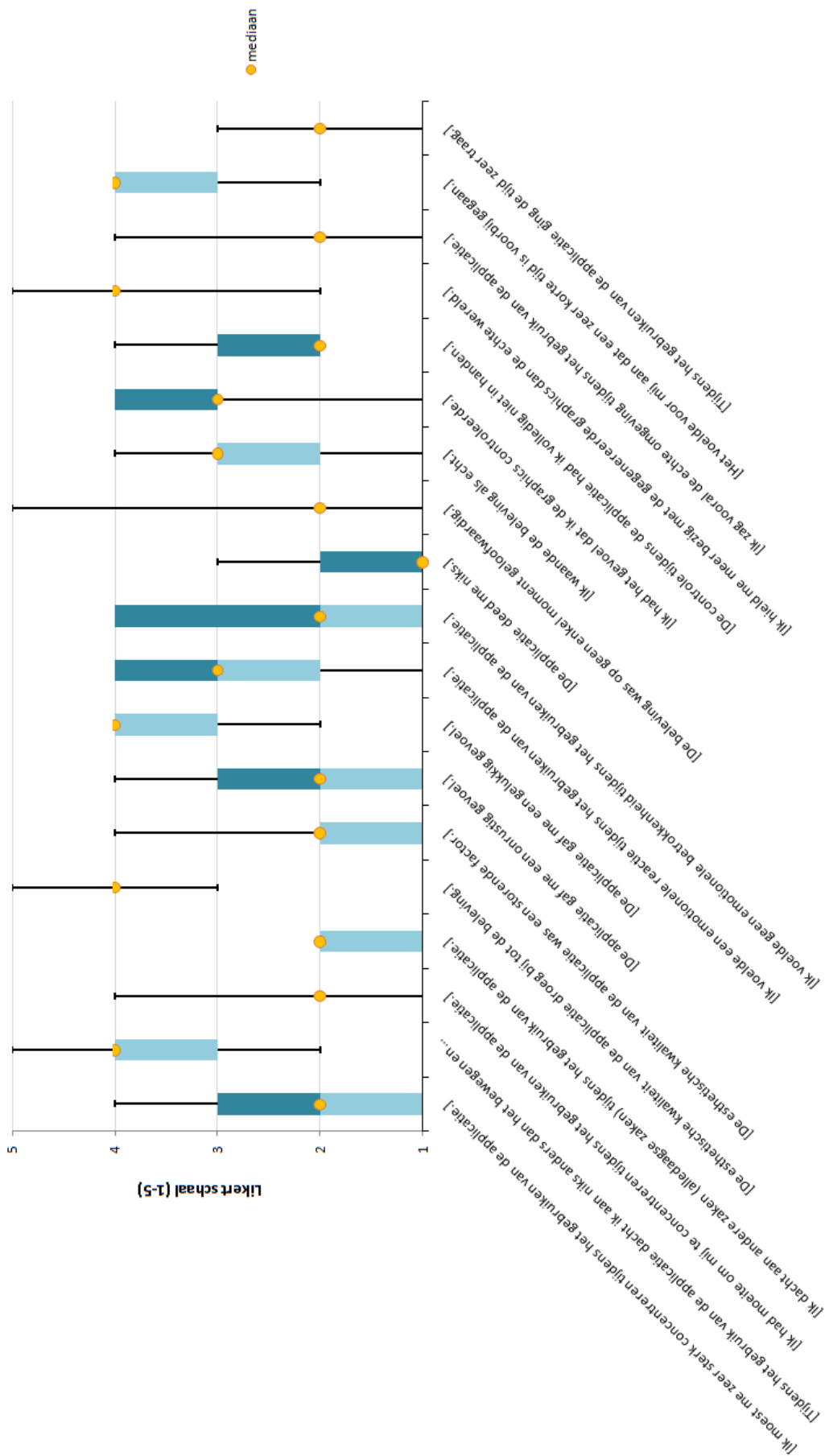
19. Tijdens het gebruiken van de applicatie ging de tijd zeer traag.

Bijlage C

Boxplotgrafiek van resultaten uit eerste evaluatiesessie

Deze boxplotgrafiek toont de resultaten van de questionnaire uit de eerste evaluatiesessie. Elke vraag van de questionnaire werd hierin opgenomen. Op de verticale as zien we de score op een Likert schaal (1 tot 5), op de horizontale as zijn alle vragen opgesomd.

BIJLAGE C. BOXPLOTGRAFIEK VAN RESULTATEN UIT EERSTE EVALUATIESESSIE 80

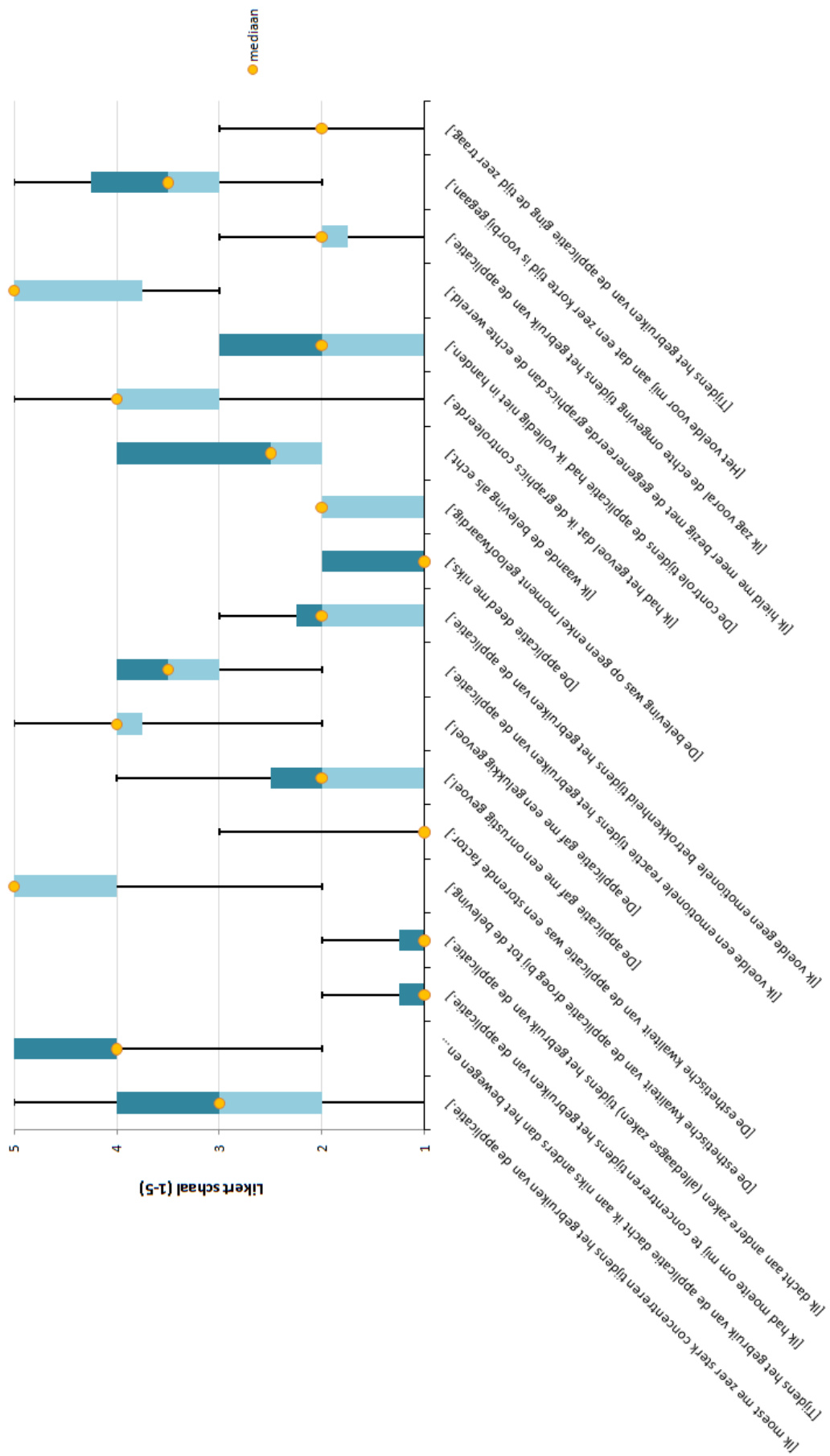


Bijlage D

Boxplotgrafiek van resultaten uit tweede evaluatiesessie

Deze boxplotgrafiek toont de resultaten van de questionnaire uit de tweede evaluatiesessie. Elke vraag van de questionnaire werd hierin opgenomen. Op de verticale as zien we de score op een Likert schaal (1 tot 5), op de horizontale as zijn alle vragen opgesomd.

BIJLAGE D. BOXPLOTGRAFIEK VAN RESULTATEN UIT TWEEDE EVALUATIESESSIE82



Bijlage E

Poster

Virtual LSD

Audio visual immersion

Augment the environment of a user in real-time using sound data

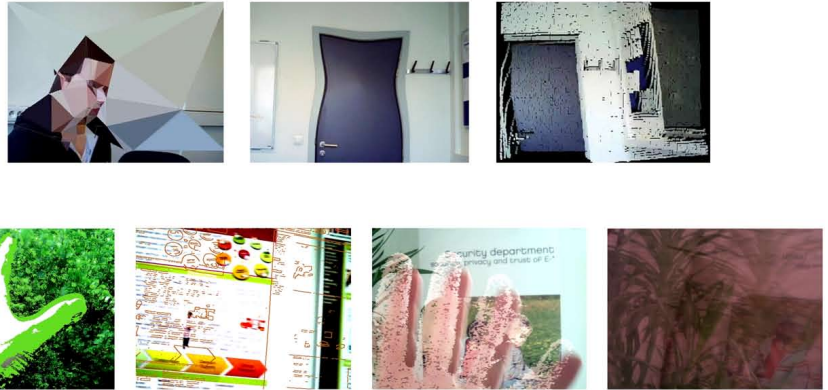
Idea

The user has an augmented reality goggle attached to a sound-playing device (iPhone, Laptop, mp3 player).

While (s)he's walking around town, various illustrations are generated in real-time based on the music (s)he hears.



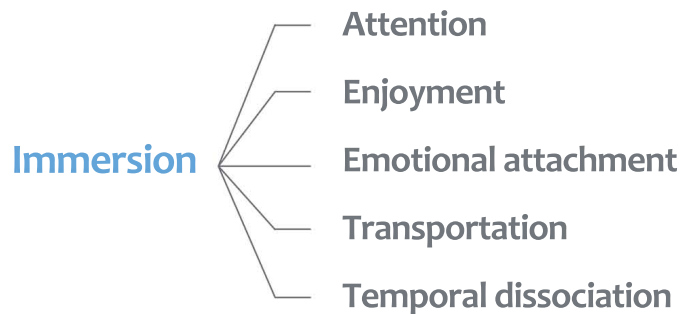
Footage



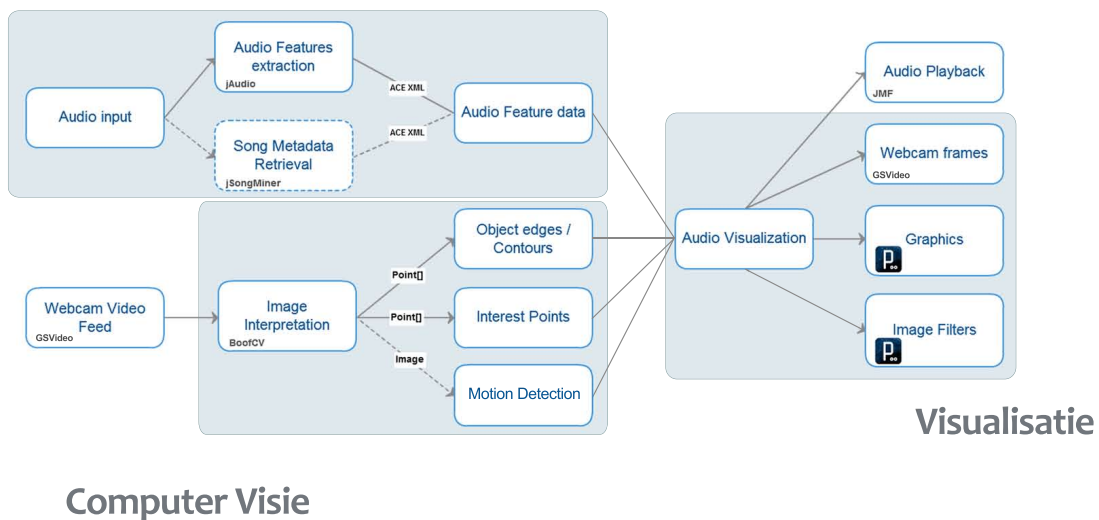
Evaluation

In this thesis, the experience and immersion of the user is studied. Different ways of evaluations are used to explore the effect of the Virtual LSD application on the music experience of the user.

A questionnaire was developed to evaluate immersion, based on 5 key components; attention, enjoyment, emotional attachment, transportation and temporal dissociation.



Music Information Retrieval



Bibliografie

- [1] R. AGARWAL AND E. KARAHANNA, *Time flies when you're having fun: Cognitive absorption and beliefs about information technology usage*, MIS Quarterly, 24 (2000), pp. 665–694.
- [2] R. ARNHEIM, *New Essays on the Psychology of Art*, University of California Press, 1986.
- [3] I. BIEDERMAN, *Recognition-by-components: a theory of human image understanding*, Psychol Review, 94 (1987), pp. 115–147.
- [4] R. BRESIN, *What is the color of that music performance*, 2005, pp. 367–370.
- [5] E. BROWN AND P. CAIRNS, *A grounded investigation of game immersion*, in CHI '04 extended abstracts on Human factors in computing systems, CHI EA '04, New York, NY, USA, 2004, ACM, pp. 1297–1300.
- [6] N. BURTONYK AND M. WEIN, *Interactive skeleton techniques for enhancing motion dynamics in key frame animation*, in Proceedings of the 3rd annual conference on Computer graphics and interactive techniques, SIGGRAPH '76, New York, NY, USA, 1976, ACM, pp. 51–51.
- [7] J. CANNY, *A computational approach to edge detection*, IEEE Trans. Pattern Anal. Mach. Intell., 8 (1986), pp. 679–698.
- [8] M. K. D. COOMANS AND H. J. P. TIMMERMANS, *Towards a taxonomy of virtual reality user interfaces*, in Proceedings of the IEEE Conference on Information Visualisation, IV '97, IEEE Computer Society, 1997, pp. 27–29.
- [9] CREATIVE RESEARCH SYSTEMS, *Survey design*. <http://www.surveysystem.com/sdesign.htm>, June 2012.
- [10] M. CSIKSZENTMIHALYI, *Flow: The Psychology of Optimal Experience*, Harper Perennial, March 1991.
- [11] L. L. CUDDY, *The color of melody*, Music Perception: An Interdisciplinary Journal, 2 (1985), pp. pp. 345–360.

- [12] C. DAVIES AND J. HARRISON, *Osmose: towards broadening the aesthetics of virtual reality*, SIGGRAPH Comput. Graph., 30 (1996), pp. 25–28.
- [13] T. DOLL, R. MIGNECO, J. SCOTT, AND Y. KIM, *An audio dsp toolkit for rapid application development in flash*, in Multimedia Signal Processing, 2009. MMSP '09. IEEE International Workshop on, oct. 2009, pp. 1–6.
- [14] J. S. DOWNIE, *Music information retrieval.*, ARIST, (2003), pp. 295–340.
- [15] C. T. EAGLE, *Music and lsd: An empirical study.*, Journal of Music Therapy, 9 (1972), pp. 23–36.
- [16] V. FERRARI, T. TUYTELAARS, AND L. V. GOOL, *Markerless augmented reality with a real-time affine region tracker*, in Procs. of the IEEE and ACM Intl. Symposium on Augmented Reality, vol. I, IEEE Computer Society, October 2001, pp. 87–96.
- [17] J. GAY, *The history of flash.* http://www.adobe.com/macromedia/events/john_gay/page04.html, June 2012.
- [18] M. HALL, E. FRANK, G. HOLMES, B. PFAHRINGER, P. REUTEMANN, AND I. H. WITTEN, *The weka data mining software: an update*, SIGKDD Explor. Newsl., 11 (2009), pp. 10–18.
- [19] A. HOFMANN, *LSD My problem child: reflection on sacred drugs, mysticism and science*, MAPS, 2005.
- [20] J. A. JACKO AND A. SEARS, eds., *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2003.
- [21] S. JAMIESON, *Likert scales: how to (ab)use them.*, Medical Education, 38 (2004), pp. 1217–1218.
- [22] C. JENNETT, A. L. COX, P. CAIRNS, S. DHOPAREE, A. EPPS, T. TIJS, AND A. WALTON, *Measuring and defining the experience of immersion in games*, Int. J. Hum.-Comput. Stud., 66 (2008), pp. 641–661.
- [23] P. JUSLIN, *Music and Emotion: Theory and Research (Series in Affective Science)*, Oxford University Press, Nov. 2001.
- [24] R. KOSARA, *Visualization criticism - the missing link between information visualization and art*, in Proceedings of the 11th International Conference Information Visualization, IV '07, Washington, DC, USA, 2007, IEEE Computer Society, pp. 631–636.
- [25] O. LARTILLOT AND P. TOIVIAINEN, *A matlab toolbox for musical feature extraction from audio*, in Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07), Bordeaux, France, September 10-15, 2007, 2007.

- [26] C. LAURIER, P. HERRERA, M. MANDEL, AND D. ELLIS, *Audio Music Mood Classification using Support Vector Machine*, 2007, pp. 2–4.
- [27] C. LAURIER, O. LARTILLOT, T. EEROLA, AND P. TOIVIAINEN, *Exploring relationships between audio features and emotion in music*, in Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music, European Society for the Cognitive Sciences of Music, 2009.
- [28] Z. LIEBERMAN, T. WATSON, AND A. CASTRO, *Openframeworks*. <http://www.openframeworks.cc/about/>, June 2012.
- [29] T. LINDBERG, *Edge detection and ridge detection with automatic scale selection*, in Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96), CVPR '96, Washington, DC, USA, 1996, IEEE Computer Society, pp. 465–470.
- [30] K. D. MARTIN, E. D. SCHEIRER, AND B. L. VERCOE, *Music content analysis through models of audition*, in Proc.ACM Multimedia Workshop on Content-Based Processing of Music, 1998.
- [31] B. MATHIEU, S. ESSID, T. FILLON, J. PRADO, AND G. RICHARD, *Yaafe, an easy to use and efficient audio feature extraction software*, in 11th International Society for Music Information Retrieval Conference (ISMIR 2010), 210, pp. 441–446.
- [32] S. MAY, *Sony hmz-t1 personal 3d viewer*. <http://reviews.cnet.co.uk/tv-accessories/sony-hmz-t1-personal-3d-viewer-review-50005000/>, Apr. 2012.
- [33] D. MCENNIS, C. MCKAY, AND I. FUJINAGA, *Jaudio: A feature extraction library*, in International Conference on Music Information Retrieval, 2005.
- [34] C. MCKAY, *Automatic Music Classification with jMIR*, PhD thesis, McGill University, Montreal, 2010.
- [35] I. MIERSWA AND K. MORIK, *Automatic feature extraction for classifying audio data*, Machine Learning, 58 (2005), pp. 127–149.
- [36] R. MIGNECO, T. DOLL, J. SCOTT, C. HAHN, P. DIEFENBACH, AND Y. KIM, *An audio processing library for game development in flash*, in Games Innovations Conference, 2009. ICE-GIC 2009. International IEEE Consumer Electronics Society's, aug. 2009, pp. 201 –209.
- [37] D. MITROVI, M. ZEPPELZAUER, AND C. BREITENEDER, *Chapter 3 - features for content-based audio retrieval*, in Advances in Computers: Improving the Web, M. V. Zelkowitz, ed., vol. 78 of Advances in Computers, Elsevier, 2010, pp. 71 – 150.

- [38] T. MOESLUND, *Image and Video Processing*, Computer Vision and Media Technology, Aalborg University, 2008.
- [39] J. NIELSEN, *Usability Engineering*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [40] B. PARVIZ, S. LEE, AND S. THRUN, *Project glass*. <https://plus.google.com/111626127367496192147/posts>, Apr. 2012.
- [41] M. M. RUXANDA, B. Y. CHUA, A. NANOPOULOS, AND C. S. JENSEN, *Emotion-based music retrieval on a well-reduced audio feature space*, in Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '09, Washington, DC, USA, 2009, IEEE Computer Society, pp. 181–184.
- [42] E. D. SCHEIRER, *Bregman's chimerae: Music perception as auditory scene analysis*, in In Proc. Int. Conf. on Music Perception and Cognition, 1996.
- [43] E. SCHUBERT, *Update of the hevner adjective checklist.*, Perceptual and motor skills, 96 (2003), pp. 1117–1122.
- [44] J. J. SCOTT, R. MIGNECO, B. G. MORTON, C. M. HAHN, P. DIEFENBACH, AND Y. E. KIM, *An audio processing library for mir application development in flash.*, in ISMIR, J. S. Downie and R. C. Veltkamp, eds., International Society for Music Information Retrieval, 2010, pp. 643–648.
- [45] J. SEO, *Aesthetics of Immersion in Interactive Immersive Environments: A Phenomenological Case Study of Light Strings*, LAP Lambert Acad. Publ., 2011.
- [46] D. SHIFFMAN, *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*, The Morgan Kaufmann Series in Computer Graphics, Morgan Kaufmann/Elsevier, 2008.
- [47] B. SHNEIDERMAN AND C. PLAISANT, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Pearson Addison Wesley, 2005, ch. 14.
- [48] T. SIMONITE, *At ces, a preview of tomorrow's wearable computers*. <http://www.technologyreview.com/news/426621/at-ces-a-preview-of-tomorrows-wearable-computers/>, Apr. 2012.
- [49] M. SLATER, *Measuring presence: A response to the witmer and singer presence questionnaire*, Presence: Teleoper. Virtual Environ., 8 (1999), pp. 560–565.
- [50] M. SLATER AND M. USOH, *Body centred interaction in immersive virtual environments*, in Artificial Life and Virtual Reality, John Wiley and Sons, 1994, pp. 125–148.
- [51] I. E. SUTHERLAND, *A head-mounted three dimensional display*, in Proceedings of the December 9-11, 1968, fall joint computer conference, part I, AFIPS '68 (Fall, part I), New York, NY, USA, 1968, ACM, pp. 757–764.

- [52] Y. TAKEUCHI AND K. PERLIN, *Clayvision: the (elastic) image of the city*, in Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems, CHI '12, New York, NY, USA, 2012, ACM, pp. 2411–2420.
- [53] Z. TU, X. CHEN, A. L. YUILLE, AND S.-C. ZHU, *Image parsing: Unifying segmentation, detection, and recognition*, in Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, ICCV '03, Washington, DC, USA, 2003, IEEE Computer Society, pp. 18–25.
- [54] T. TUYTELAARS AND K. MIKOLAJCZYK, *Local invariant feature detectors: a survey*, Found. Trends. Comput. Graph. Vis., 3 (2008), pp. 177–280.
- [55] G. TZANETAKIS AND P. COOK, *Marsyas: a framework for audio analysis*, Org. Sound, 4 (1999), pp. 169–175.
- [56] D. W. F. VAN KREVELEN AND R. POELMAN, *A Survey of Augmented Reality Technologies, Applications and Limitations*, The International Journal of Virtual Reality, 9 (2010), pp. 1–20.
- [57] S. VIEILLARD, I. PERETZ, N. GOSSELIN, S. KHALFA, L. GAGNON, AND B. BOUCHARD, *Happy, sad, scary and peaceful musical excerpts for research on emotions*, Cognition and Emotion, 22 (2008), pp. 720–752.
- [58] B. G. WITMER AND M. J. SINGER, *Measuring presence in virtual environments: A presence questionnaire*, Presence: Teleoper. Virtual Environ., 7 (1998), pp. 225–240.
- [59] P. L. WORTHINGTON, *Enhanced canny edge detection using curvature consistency*, in Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1 - Volume 1, ICPR '02, Washington, DC, USA, 2002, IEEE Computer Society, pp. 596–599.
- [60] P. ZAHORIK AND R. L. JENISON, *Presence as being-in-the-world*, Presence: Teleoper. Virtual Environ., 7 (1998), pp. 78–89.