Designing interaction for browsing media collections (by similarity)

Christian Frisson
CHRISTIAN FRISON

DESIGNING INTERACTION FOR
BROWSING MEDIA COLLECTIONS
(BY SIMILARITY)

DISSERTATION SUBMITTED
TO THE FACULTY OF ENGINEERING OF THE UNIVERSITY OF MONS,
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN APPLIED SCIENCE

JURY:
THIERRY DUTOIT (UNIVERSITY OF MONS) – SUPERVISOR
STÉPHANE DUPONT (UNIVERSITY OF MONS)
XAVIER SIEBERT (UNIVERSITY OF MONS)
JEAN VANDERDONCKT (CATHOLIC UNIVERSITY OF LOUVAIN)
MARCELO WANDERLEY (MCGILL UNIVERSITY)
JEF WIJSEN (UNIVERSITY OF MONS)

UNIVERSITY OF MONS, NUMEDIART INSTITUTE, TCTS LAB
Abstract

Sound designers source sounds in massive and heavily tagged collections. When searching for media content, once queries are filtered by keywords, hundreds of items need to be reviewed. How can we present these results efficiently?

This doctoral work aims at improving the usability of browsers of media collections by blending techniques from multimedia information retrieval (MIR) and human-computer interaction (HCI). We produced an in-depth state-of-the-art on media browsers. We overviewed HCI and MIR techniques that support our work: organization by content-based similarity (MIR), information visualization and gestural interaction (HCI). We developed the MediaCycle framework for organization by content-based similarity and the DeviceCycle toolbox for rapid prototyping of gestural interaction, both facilitated the design of several media browsers. We evaluated the usability of some of our media browsers.

Our main contribution is AudioMetro, an interactive visualization of sound collections. Sounds are represented by content-based glyphs, mapping perceptual sharpness (audio) to brightness and contour (visual). These glyphs are positioned in a starfield display using Student t-distributed Stochastic Neighbor Embedding (t-SNE) for dimension reduction, then a proximity grid optimized for preserving direct neighbors. Known-item search evaluation shows that our technique significantly outperforms a grid of sounds represented by dots and ordered by filename.
Acknowledgements

A thesis stands for nothing without proper acknowledgements. I chose to start with people that are directly concerned with this work or indirectly enabled it (the jury of this thesis, my colleagues, research friends, sources of fundings, opensource software makers, collaborators), and gradually reach people that I know or know me the most (friends and family).

Jury

I owe my sincere gratitude to Thierry Dutoit (University of Mons, Belgium) for having accepted to supervise my thesis, against all initial odds, after I had been dismissed from a first attempt at another university. He also ensured my sustained comfortable research assistantship contracts throughout this work. I hope that this manuscript proves that I was right to insist and convince him. I will also remember him as role model in emotional control: I have never seen him angry (corridor gossip discussions maintain the legend of one single event), eventhough one could consider that I tried hard to make him so, for instance through some of my "should I hit send?" emails.

I am redeemable to Stéphane Dupont and Xavier Siebert (both: University of Mons, Belgium), who as regard to this thesis founded the MediaCycle framework, and for coping with my stubborn inclination in approaching our research tracks more from a human-computer interaction perspective rather than a multimedia information retrieval strategy.

I am endowed to Jean Vanderdonckt (Université catholique de Louvain, Belgium) for having recommended me references and books on information visualization, human-computer interaction and usability evaluation, the research fields framing this thesis that drive my interest the most. After all, the book Readings In Information Visualization that he advised me to read years before (what I skipped at the time), is a hub to references that I stumbled upon later.

I thank Marcelo Wanderley (McGill University, Canada) and Jeff Wijsen (University of Mons, Belgium) for having accepted to be part of my jury including proofreading this manuscript and evaluating my thesis defense. Marcelo Wanderley opened my mind on gestural control (for musical applications, with tactilo-proprio-kinesthetic feedback) during my Master thesis together with Claude Cadoz, and we could discuss more recently about the methodology for the design and evaluation of such interaction techniques. Jef Wijsen spotted all my overly long sentences, inaccuracies about clustering, and explanations of scientific directions that lacked transparency.
Colleagues

My “tangible” prototyping efforts with the MediaCycle framework feature some work of colleagues from UMONS who participated to its development: Laurent Couvreur, Stéphane Dupont, Alexis Moinet, Thierry Ravet, Xavier Siebert, Damien Tardieu, Jérôme Urbain.

Evaluating some of these prototypes has been greatly enabled by Nicolas Riche and Willy Yvart, they discussed the experimental designs and facilitated the user tests at UMONS and Université de Valenciennes et du Hainaut Cambrésis (UVHC). I also thank the hundred testers from these universities that donated their time without significant compensation.

I feel gratitude for Nicolas Riche and Matthieu Duvinage for our thesis-related discussions, from the choice of test statistics to the “bottom wave” phase of the PhD and life after.

Damien Tardieu came up with the idea that hooking up a depth sensing camera to the AudioCycle application would result in a demo for exploring sounds on a map by navigating physically over the space, targeted for novice users, beyond multimedia information retrieval specialists and sound experts. It became LoopJam.

I keep good memories of the AudioGarden brainstorming and prototyping sessions with Cécile Picard-Limpens and Damien Tardieu, around Parvis de St Gilles.

I has been a nice experience to work with Fabien Grisard on his Master Thesis at the numediart Institute. He was at the the time student from the “Art, Science, Technology” Master from INP Grenoble and ACROE, where I studied years before.

I am redeemable to Todor Todoroff notably for having pitched LoopJam to be installed at the Musical Instruments Museum in Brussels during La Semaine du Son in 2013, and for discussions without compromises about joint numediart trimestrial projects.

It has been a pleasure to improvize these BandCycle jams with Onur Babacan, blending eastern and western musical influences with acoustic and digital instruments. During these soundful getaways, some discussions popped up and lead us to also blend our work efforts on our MakamCycle project on Turkish Makam music analysis with MediaCycle.

Computer-related logistics have been seamlessly maintained and managed by William Van Hoeck who was always there to help install useful applications for our workflow.

I thank Johan Dechristophoris for having built a wooden stand holding hardware parts of a Novint Falcon force-feedback 3D mouse in an attempt to repurpose it to a force-feedback rotary controller.

Nathalie Durieux has always been of great help and with great calm for dealing with administrative businesses, from hardware acquisitions to conference expenses coverage.

I much appreciated the work of Laura Colmenares Guerra who edited the numediart demo videos including the timeline of MediaCycle.

Charles-Alexandre Delestage and Willy Yvart have designed the AudioMetro poster visuals that recur along this manuscript (including the front cover).
Research friends

I am grateful to Rudi Giot (IRISIB, Belgium) for having allowed me to setup at IRISIB LARAS an experiment, Effective Affinities, that unfortunately couldn’t serve this thesis, and his colleagues Alexis Rochette and Ludovic Laffineur for having taken longitudinal known-item search tests repeated over several days.

I would like to thank Eric Schayes and Simon Uyttenhove (at the time students at IRISIB, Belgium) for pitching ideas and coming up with a late-breaking design for LavaAMP for the ACM UIST 2013 Student Innovation Contest. I must reveal that this prototype, among the all content-based browsers generated throughout this thesis, gave me the most frills.

I am grateful to Klaus Schoeffmann (Klagenfurt University, Austria) for having accepted our VideoCycle video browser as challenger in the Video Browser Showdown, introducing me to known-item search user evaluations.

I am grateful to Thomas Grill (OFAI, Austria) for our insightful conversations on content-based audio browsers, for creating and maintaining flext (PureData and Max/MSP layer) used in the DeviceCycle toolbox, for opening his work on visualization of textural sounds under a Creative Commons license what inspired an upcoming transition to MediaCycle towards web-based applications.

I would like to thank Felix Kistler (Augsburg University, Germany) for quickly answering many questions regarding the adaptation of FUBI for the MashtaCycle project.

I am grateful to the chairs and teachers of ACM TEI 2013 Graduate Student Consortium (GSC) and the ACM IHM 2010 doctoral consortium for their valuable evaluation of my doctoral work. The 2-hour idea pitching session through tangible poster crafting was inspiring. TEI 2013 organizers made a nice gesture by waiving the conference fees and covering all expenses for GSC participants, what was in great part handled by Tom Moher (University of Illinois at Chicago, USA) through a grant from the U.S. National Science Foundation (NSF).

Achieving a PhD thesis has been questioned over and again with past colleagues and research friends from other universities that became and remained present friends (alphabetically): Sema Alaçam (parametric architect), Katia Cánepa Vega (beauty technologist), Bruno Dumas (inputs fuser), Je(h)an-Julien Filatriau (sound texture trotter), Lander Ibarra Valle (heat transferer), Ceren Kayalar (mobile reality augmenter), Suzanne Kieffer (usability strategist), Lionel Lawson (serial prototyper), Rémy Lehembre (“Dream Machine”), Ghislain Retaureau (full throttler), Cédric Simon (tree miner), Arnaud Thabot (unstarchy cook/book expert), Doug Van Nort (music grapher)…
Funders

I am redeemable to Guy Vanden Bemden for driving 3 successful projects answering the GreenTIC call from the Walloon Region of Belgium, one of which named SonixTrip I was chosen to work on, before he left to work for Belgian post-production studios Dame Blanche. Didier Lefebvre (Océ Software Laboratories, Belgium) coordinated the grant proposal for the SonixTrip project.

I thank the Walloon Region for funding the numediart research program (2007-2013) that was later morphed into the numediart Institute, and the SonixTrip project through the GreenTIC call (2013-2016), providing a full financial coverage for this thesis.

Opensource software makers

During my graduate studies I became passionate with multitrack audio recording and computer music. Switching to Linux at that time turned my creative endeavors into sessions of dissections of opensource (audio) software. Recompiling applications to install bleeding-edge updates and kernels for realtime preemption greatly facilitated the transition towards prototyping as approached throughout this thesis. I am redeemable to opensource software makers from various projects that sped up the development of MediaCycle (alphabetically): CMake, DokuWiki, ffmpeg, flex, Google Breakpad, hidapi, liblo, libnifalcon, MacPorts, Mercurial, OpenAL (Soft), Octave, OpenCV, OpenSceneGraph, PureData, Qt, STK, R, YAAFE. I am also redeemable to opensource software makers of applications that have been feeding my creativity ever since: Mixxx and SooperLooper. I hope that I can contribute back in the future in some sort of way.
Collaborators

I thank Etienne Dontaine and Matthieu Michaux from post-production studios Dame Blanche for our meetings and discussions that taught us a lot about sound design and opened opportunities for collaborations.

I thank Guillaume Chappez, Rudi Giot, Gauthier Keyaerts, Julien Poidevin, Jean-Louis Poliart, Laszlo Umbreit for having accepted to compose audio loop collections for Loopjam installed in the underground hall of movie theater Galeries for the Citysonic #10 festival, in Brussels, Belgium, from September 27 to October 14 2012. The best collateral way to celebrate one’s 30th birthday through an art exhibition opening.

I thank Laura Colmenares Guerra, Bénédicte and Laure-Anne Jacobs aka Larbits Sisters and Vicki Bennett aka People Like Us for our interesting discussions about video browsing and organization (and VideoCycle).

I thank Sébastien Biset and Pierre Hemptinne from La Médiathèque ASBL (later renamed PointCulture) for discussions about the organization of multimedia content, particularly through their Archipel physical/online browsers curating collections of “unclassifiable” music.

I thank Philippe Franck, Lucie Knockaert, Emilien Baudelot and the team of Transcultures ASBL for curating our interactive sound installations Loopjam and MashtaCycle to local and remote audiences through the Transnumériques #4 (2012) and Citysonic #10 (2012) and Citysonic #13 (2013) festivals.

I thank Marianne Binard, Stephan Dunkelman and the organizing team of “La Semaine du Son” and Wim Verhulst from the Musical Instruments Museum of Brussels for having hosted Loopjam in 2013.

I thank Carla Scaletti from Symbolic Sound and Rudi Giot from IRISIB, organizers of KISS 2013, for having accepted The Listening Room (forking MashtaCycle) to be an interactive jukebox for sound collections composed with Kyma, composed by Ilker Isikyakar, Gauthier Keyaerts, Olga Oseth, Carla Scaletti.
Friends

I am redeemable to Christophe, Thierry and Vincent, media librarians at La Médiathèque from Nilvange, France, during my childhood, who I consider having acted as music/movie mentors, complementarily through their favorite genres.

My Turkish friends have been welcoming me numerous times mostly in Istanbul and Izmir for distant working and leisure, alphabetically: Burcu Akyol Ersoy, Sema Alaçam, Elvin Erkut, Oya Gürsoy Yılmaz, Şerife Kapukaya Isbilir, Ceren Kayalar, Başak Sözer Elgin...

To Alain Labouverie, Aurélie Moulaert, Aline Peronne, Cédric Simon; I hope that you now understand my sudden cancellation due to this work of the 2013 ski holidays that I would have enjoyed sharing once more with you. I would like to thank Sophie Moreau and Fred Demilito for our transcendent getaways during this thesis. I’ll keep on enjoying weekly meetings with Bruno Dumas, Suzanne Kieffer, and Cédric Simon. I wish I could do the same with remote Ghislain Retaureau and Yves Desnos, beyond our PhD experiences.

Family

I am sure that the life and career choices of my parents Carmen Frisson, a socializing nurse and perfect cook, and Patrice Frisson, a civil engineering urban drawings drafter and bass guitar player, influenced mine subconsciously. They were always there to facilitate and acknowledge my choices.

I am eager to go ski again with my brother Michel and his family, now that we have each passed one major life milestone. I had to complete this thesis to understand it properly.

I am glad that my grandparents Frida and Albert Frisson could attend the public presentation of this thesis and celebrate thereafter. Maria and Charles Strubel would have certainly been also glad to live this.

I am lucky to have plenty of cousins and aunts/uncles. I thank Stéphanie Strubel, Jacques Pedrini, and Leonor Pedrini for always attending celebrations of milestones of my life without hesitation. I thank Alex and Nicolas Strubel, and their parents Evelyne and Pascal, for our downhill/mountain bike trips that would efficiently reset my mind. I thank Elise and Estelle Vagost, and their parents Marie-Thérèse and Daniel, for having also attended to my public presentation. I still need to hand you back the Texas Instruments calculators. I thank Brigitte and Alain Giacometti, and their daughters Laura and Sandra, for our family reunions in which my thesis was (thankfully) off-topic. My British cousins Romy and Simon Strubel contributed greatly in my enjoyment of learning English, some of their words must be scattered in this thesis.

I owe Elise Huwart for having taught me how to breathe correctly before the first oral presentation of mine that I found acceptable. Let’s chill out together behind the scenes now!
# Contents

1. **Introduction: context, terminology, concerns, statement**  
   25

2. **Background: browsing media content (by similarity)**  
   43

3. **Method: interaction, organization**  
   107

4. **Design, prototypes, implementation**  
   133

5. **Experiments, evaluations**  
   183

6. **Conclusion**  
   247

7. **References**  
   253
List of Figures

1.1 *The Disciplines of User Experience Design*, forked from Envis Precisely – 28
1.2 Word cloud of the thesis references containing the word “browser” 29

2.1 Screenshot of *Web Forager* by Card et al. (1996) 46
2.2 Screenshot of *Raskin* file browser (2001) 47
2.3 Screenshot of *BumpTop* by Anand Agarawala (2006) 48
2.4 Screenshot of *Liquifile* by Waldeck et al. (2004) 49
2.5 Screenshot of *Film Finder* by Ahlberg & Shneiderman (1993) 50
2.6 Screenshots of *AAU Video Browser* by Del Fabro et al. (2010) 51
2.7 Screenshot of *Panopticon* by Jackson et al. (2013) 52
2.8 Screenshot of *SonicBrowser* by Fernström and Brazil (2001) 53
2.9 Screenshots of visualizations of music libraries by Torrens et al. (2004) 54
2.10 Screenshot of *The amblr* by Stewart et al. (2008) 55
2.11 Screenshot of *AudioFinder* by Iced Audio (1986) 56
2.12 Screenshot of *Soundminer HD* (2002) featuring its *Launchpad* view 57
2.13 Screenshot of *Adobe Bridge* (2005) 58
2.14 Picture of *Vintage Radio Interface* by Hopmann et al. (2012) 59
2.15 Screenshot of *Greenpoint* by Witten et al. (2000) 60
2.16 Screenshot of *Bohemian Bookshelf* by Thudt et al. (2012) 61
2.17 Picture of *SpeechSkimmer* by Barry Arons (1993) 62
2.18 Screenshots of *SoundFisher* by Muscle Fish (1996) 63
2.19 Picture of *Marsyas3D* by Tzanetakis et al. (2001) 64
2.20 Picture of *SmartMusicKIOSK* by Goto (2003) 65
2.21 Screenshot of *Catark* by Schwarz et al. (2004) 66
2.22 Screenshots of *Musicream* by Goto and Goto (2005) 67
2.23 Screenshot of *Musicminer* by Moerchen et al. (2005) 68
2.24 Screenshot of *Search Inside the Music* by Lamere & Eck (2007) 69
2.25 Pictures of *MusicRainbow* by Pampalk and Goto (2006) 70
2.26 Screenshot of *Globe Of Music* by Leitich and Topf (2007) 71
2.27 Screenshot of *MusicBox* by Anita Lillie (2008) 72
2.28 Screenshots of *soniXplorer* by Lübbers and Jarke (2009) 73
2.29 Screenshot of SoundTorch by Heise et al. (2008) 74
2.30 Screenshot of SongExplorer by Julià et al. (2009) 75
2.31 Screenshot of MuVis by Dias et al. (2010) 76
2.32 Mockup of Sonarflow by Spectralminds (2010) 77
2.33 Screenshot of MusicGalaxy by Stober et al. (2010) 78
2.34 Screenshots of Informedia Digital Library Interface by Christel et al. (1995-2008) 79
2.35 Screenshots of FutureViewer by Campanella et al. (2005) 80
2.36 Annotated screenshot of VideoSOM by Bärecke et al. (2006) 81
2.37 Screenshot of ITEC Video Explorer by Schoeffmann et al. (2010) 82
2.38 Screenshots of MediaMill by de Rooij et al. (2010) 83
2.39 Screenshots of MediaTable by de Rooij et al. (2010) 84
2.40 Screenshot of Galaxy Browser by Pang et al. (2011) 85
2.41 Screenshot of Joanneum Video Browser by Bailer et al. (2012) 86
2.42 Screenshot of OVIDIUS platform by Bursuc et al. (2012) 87
2.43 Screenshot of 3D Thumbnail Ring by Schoeffmann et al. (2012) 88
2.44 Screenshot of Graphic Object Searcher by Ventura et al. (2012) 89
2.45 Screenshot of HiStory by Hürst and Darzentas (2012) 90
2.46 Screenshot of 3D Filmstrip by Hudelist et al. (2013) 91
2.47 Picture of Video Archive Explorer by Haesen et al. (2013) 92
2.48 Screenshot of Sonic Mapper by Scavone et al. (2002) 93
2.49 Picture of Pockets Full Of Memories by George Legrady (2001) 94
2.50 Screenshots of Cell Tango by George Legrady (2006-2010) 95
2.51 Picture of Shape of Song by Martin Wattenberg (2006) 96
2.52 Picture of Every Shot, Every Episode (2001) 97
2.53 Screenshot of cinemetrics by Frederic Brodbeck (2011) 98

3.1 Content-based dataflow 108
3.2 Graphical representation of common metrics 111
3.3 A state-of-the-art visualization for waveform skimming 115
3.4 AudioFish waveform display in Mixxx 119
3.5 Visual summaries of Blender movie Elephants Dreams 122
3.6 Photo browser views by Hudelist et al. (2013) 124

4.1 MediaCycle-related numediart projects clustered by media type 135
4.2 MediaCycle dataflow 136
4.3 MediaCycle file tree 137
4.4 MediaCycle classes: media types 138
4.5 MediaCycle classes: media data 139
4.6 **MediaCycle** classes: base plugins

4.7 **MediaCycle** classes: plugin parameters

4.8 **MediaCycle** classes: plugin libraries

4.9 **MediaCycle** classes: media library plugins

4.10 **MediaCycle** classes: media reader plugins

4.11 **MediaCycle** classes: media features plugins

4.12 **MediaCycle** classes: timed-features plugins

4.13 **MediaCycle** classes: segmentation plugins

4.14 **MediaCycle** classes: thumbnailer plugins

4.15 **MediaCycle** classes: cluster method plugins

4.16 **MediaCycle** classes: neighbor method plugins

4.17 **MediaCycle** classes: cluster position plugins

4.18 **MediaCycle** classes: neighbor positions plugins

4.19 **MediaCycle** classes: media renderer plugins

4.20 **MediaCycle** classes: client plugins

4.21 **DeviceCycle**: 3dconnexion Space Navigator 3D mouse

4.22 **DeviceCycle**: Contour Design Shuttle Pro 2 jog wheel

4.23 **DeviceCycle**: Novint Falcon 3DOF force-feedback “mouse”

4.24 **DeviceCycle**: Apple multitouch trackpads

4.25 numediart projects and **MediaCycle**

4.26 **AudioCycle**: screenshot of the initial version (2008)

4.27 **AudioCycle**: bimanual media browsing

4.28 **AudioGarden**: screenshot with the Gramophone layout (2010)

4.29 **AudioGarden**: screenshot with the Flower layout (2010)

4.30 **LoopJam**: exhibition in Seneffe, Belgium (2011) © numediart

4.31 **MashtaCycle**: prototyping (2013) © Laura Colmenares Guerra

4.32 **MashtaCycle**: gesture-sound mapping (2013) © Fabien Grisard

4.33 **The Listening Room**: installed at KISS 2013 © KISS 2013

4.34 **VideoCycle**: intra-media browsing (2012)

4.35 **VideoCycle**: Video Browser Showdown version (2012)

4.36 **LavaAMP**: video excerpt (2013) © Eric Schayes

4.37 **LavaAMP**: PumpSpark Fountain Development Kit © Microsoft

4.38 **LavaAMP**: PumpSpark Control Board © Microsoft

4.39 **LavaAMP**: team © Simon Uyttenhove

4.40 **The Grayfish Squidget**: demo at ACM TEI 2014 © Sema Alaçam

4.41 **The Grayfish Squidget** versus a jog wheel © Willy Yvart

4.42 **The Grayfish Squidget**: Gray code printout © Willy Yvart

4.43 **The Starfish eNTERFACE**: demo at ACM TEI 2014 © Sema Alaçam
4.44 Starfish eNTERFACE: clamped in upright position © Willy Yvart 174
4.45 Starfish eNTERFACE: fitted with a mouse © Willy Yvart 175
4.46 AudioMetro: concept (2014) 176
4.47 AudioMetro: content-based glyphs 176
4.48 AudioMetro: discretizing plot coordinates using a proximity grid (2014) 177
4.49 Proximity grid spiral search methods - © Wojciech Basalaj 178
4.50 AudioMetro: proportion of direct neighbors vs proximity grid sides 179

5.1 A sound designer describing his practices in his workspace 189
5.2 Search results in a sound browser for sound design – © Matthieu Michaux 191
5.3 Setting of the Video Browser Showdown (VBS) in 2013 – © MMM 2013 196
5.4 Picture of the VBS server displayed on a large TV screen – © Klaus Schoeffmann 196
5.5 Screenshot of the VideoCycle version that contested at the VBS 197
5.6 Total scores per browser at the VBS 2013 199
5.7 Successful/failed submissions per browser and run at the VBS 2013 199
5.8 Success times per browser and run at the VBS 2013 200
5.9 Tagcloud of filename terms in the OLPC library 203
5.10 Visual organization and target location for each test 1 task 207
5.11 Screenshots of the user interface for the first grid and cloud tasks 207
5.12 Histograms of scores per user with the grid and cloud views for test 1 208
5.13 Quantile plots of scores per user with the grid and cloud views for test 1 209
5.14 Success times per view (seconds) for test 1 210
5.15 Stumble times per view (seconds) for test 1 211
5.16 Recollection times per view (seconds) for test 1 211
5.17 Distances per view (in number of grid rows/columns) for test 1 212
5.18 Speeds per view (in grid rows/columns per second) for test 1 212
5.19 Discovers per view (number of unique sounds browsed per task) for test 1 213
5.20 Hovers per view (number of cumulated sounds browsed per task) for test 1 213
5.21 Top-ten shortest successful mouse paths for each task of test 1 215
5.22 Success times per grid (left) and cloud (right) tasks for test 1 216
5.23 Stumble times per grid then cloud tasks for test 1 217
5.24 Recollection times per grid then cloud tasks for test 1 217
5.25 Discovers per grid then cloud tasks for test 1 218
5.26 Hovers per grid then cloud tasks for test 1 218
5.27 Distances per grid then cloud tasks for test 1 219
5.28 Speeds per grid then cloud tasks for test 1 219
5.29 Setup of the second known-item search evaluation of sound layouts 220
5.30 Histograms of scores per user with the grid and cloud views for test 2 222
5.31 Quantile plots of scores per user with the grid and cloud views for test 2
5.32 Success times per view (seconds) for test 2
5.33 Stumble times per view (seconds) for test 2
5.34 Recollection times per view (seconds) for test 2
5.35 Distances per view (in number of grid rows/columns) for test 2
5.36 Speeds per view (in grid rows/columns per second) for test 2
5.37 Discovers per view (number of unique sounds browsed per task) for test 2
5.38 Hovers per view (number of cumulated sounds browsed per task) for test 2
5.39 User-rated evaluation of layouts as surveyed from participants to test 2
5.40 Favorite file browser layout as surveyed from participants to test 2
5.41 Usage of desktop operating systems as surveyed from participants to test 2
5.42 Setup of the third known-item search evaluation of sound layouts
5.43 Histograms of scores per user with the grid and cloud views for test 3
5.44 Quantile plots of scores per user with the grid and cloud views for test 3
5.45 Success times per view (seconds) for test 3
5.46 Stumble times per view (seconds) for test 3
5.47 Recollection times per view (seconds) for test 3
5.48 Distances per view (in number of grid rows/columns) for test 3
5.49 Speeds per view (in grid rows/columns per second) for test 3
5.50 Discovers per view (number of unique sounds browsed per task) for test 3
5.51 Hovers per view (number of cumulated sounds browsed per task) for test 3
5.52 User-rated evaluation of layouts as surveyed from participants to test 3
5.53 Favorite file browser layout as surveyed from participants to test 3
5.54 Usage of desktop operating systems as surveyed from participants to test 3
5.55 Setup of the fourth known-item search evaluation of sound layouts
5.56 Different layouts with glyphs for the same sound collection
5.57 Visual organization and target location for each task of test 1
5.58 Histograms of scores per user with the grid, album and metro views for test 4
5.59 Quantile plots of scores per user with the grid, album and metro views for test 4
5.60 Success times per view (seconds) for test 4
5.61 Stumble times per view (seconds) for test 4
5.62 Recollection times per view (seconds) for test 4
5.63 Speeds per view (in grid rows/columns per second) for test 4
5.64 Distances per view (in number of grid rows/columns) for test 4
5.65 Discovers per view (number of unique sounds browsed per task) for test 4
5.66 Hovers per view (number of cumulated sounds browsed per task) for test 4
5.67 User-rated evaluation of layouts as surveyed from participants to test 4
5.68 Favorite file browser layout as surveyed from participants to test 4
5.69 Usage of desktop operating systems as surveyed from participants to test

6.1 Cloud layout with/out high-dimensional nearest-neighbor links
List of Tables

2.1 Template of a taxonomy table to describe a media browser 44
2.2 Legend of taxonomy tables to describe a media browser 44
2.3 Taxonomy of Web Forager by Card et al. (1996) 46
2.4 Taxonomy of Raskin file browser (2001) 47
2.5 Taxonomy of BumpTop by Anand Agarawala (2006) 48
2.6 Taxonomy of Liquifile by Waldeck et al. (2004) 49
2.7 Taxonomy of Film Finder by Ahlberg & Shneiderman (1993) 50
2.8 Taxonomy of AAU Video Browser by Del Fabro et al. (2010) 51
2.9 Taxonomy of Panopticon by Jackson et al. (2013) 52
2.10 Taxonomy of SonicBrowser by Fernström and Brazil (2001) 53
2.11 Taxonomy of visualizations of music libraries by Torrens et al. (2004) 54
2.12 Taxonomy of The amblr by Stewart et al. (2008) 55
2.13 Taxonomy of AudioFinder by Iced Audio (1986) 56
2.14 Taxonomy of Soundminer HD (2002) featuring its Launchpad view 57
2.15 Taxonomy of Adobe Bridge (2005) 58
2.16 Taxonomy of Vintage Radio Interface by Hopmann et al. (2012) 59
2.17 Taxonomy of Greenstone by Witten et al. (2000) 60
2.18 Taxonomy of Bohemian Bookshelf by Thudt et al. (2012) 61
2.19 Taxonomy of SpeechSkimmer by Barry Arons (1993) 62
2.20 Taxonomy of SoundFisher by Muscle Fish (1996) 63
2.21 Taxonomy of Marsyas3D by Tzanetakis et al. (2001) 64
2.22 Taxonomy of SmartMusicKIOSK by Goto (2003) 65
2.23 Taxonomy of CataRT by Schwarz et al. (2004) 66
2.25 Taxonomy of MusicMiner by Moerchen et al. (2005) 68
2.26 Taxonomy of Search Inside the Music by Lamere & Eck (2007) 69
2.27 Taxonomy of MusicRainbow by Pampalk and Goto (2006) 70
2.28 Taxonomy of Globe Of Music by Leitich and Topf (2007) 71
2.29 Taxonomy of MusicBox by Anita Lillie (2008) 72
2.30 Taxonomy of soniXplorer by Lübbers and Jarke (2009) 73
2.31 Taxonomy of SoundTorch by Heise et al. (2008) 74
2.32 Taxonomy of SongExplorer by Julià et al. (2009) 75
2.33 Taxonomy of MuVis by Dias et al. (2010) 76
2.34 Taxonomy of Sonarflow by Spectralminds (2010) 77
2.35 Taxonomy of MusicGalaxy by Stober et al. (2010) 78
2.36 Taxonomy of Informedia Digital Library Interface by Christel et al. (1995-2008) 79
2.37 Taxonomy of FutureViewer by Campanella et al. (2005) 80
2.38 Taxonomy of VideoSOM by Bärecke et al. (2006) 81
2.39 Taxonomy of ITEC Video Explorer by Schoeffmann et al. (2010) 82
2.40 Taxonomy of MediaMill by de Rooij et al. (2010) 83
2.41 Taxonomy of MediaTable by de Rooij et al. (2010) 84
2.42 Taxonomy of Galaxy Browser by Pang et al. (2011) 85
2.43 Taxonomy of the Joanneum Video Browser by Bailer et al. (2012) 86
2.44 Taxonomy of OVIDIUS platform by Bursuc et al. (2012) 87
2.45 Taxonomy of 3D Thumbnail Ring by Schoeffmann et al. (2012) 88
2.46 Taxonomy of Graphic Object Searcher by Ventura et al. (2012) 89
2.47 Taxonomy of HiStory by Hürst and Darzentas (2012) 90
2.48 Taxonomy of 3D Filmstrip by Hudelist et al. (2013) 91
2.49 Taxonomy of Video Archive Explorer by Haesen et al. (2013) 92
2.50 Taxonomy of Sonic Mapper by Scavone et al. (2002) 93
2.51 Taxonomy of Pockets Full Of Memories by George Legrady (2001) 94
2.52 Taxonomy of Cell Tango by George Legrady (2006-2010) 95
2.53 Taxonomy of The Shape Of Song by Martin Wattenberg (2001) 96
2.54 Taxonomy of Every Shot, Every Episode by Jennifer and Kevin McCoy (2001) 97
2.55 Taxonomy of cinemetrics by Frederic Brodbeck (2011) 98
2.56 Browsers compared: organization, users, community, evaluation 100
2.57 Browsers compared: organization, visualization, interaction 102

3.1 Browsers compared: organization, visualization 113
3.2 Browsers compared: interaction 126

4.1 Taxonomy of AudioCycle (2008) 158
4.2 Taxonomy of AudioGarden file browser (2010) 160
4.3 Taxonomy of LoopJam (2011-2012) 162
4.4 Taxonomy of MashtaCycle (2013) 164
4.5 Taxonomy of The Listening Room (2013) 166
4.6 Taxonomy of VideoCycle (2013) 168
4.7 Taxonomy of LavaAMP (2013) 170
4.8 Taxonomy of the Grayfish Squidget (2014) 172
4.9 Taxonomy of the Starfish eNTERFACE (2014)  174
4.10 Taxonomy of AudioMetro (2014)  176
4.11 System-wise browsers comparison  180
4.12 User-centered browsers comparison  180

5.1 Word categories and examples sampled from the OLPC dataset  203
5.2 Summary of experimental conditions for test 1  204
5.3 Filenames of the targets for each task of test 1  207
5.4 Mann-Whitney u-tests (grid > cloud) of all variables for test 1  209
5.5 Bird-flight distances to the target for test 1  213
5.6 Paired Student t-test results comparing variables for test 1  214
5.7 Summary of experimental conditions for test 2  220
5.8 Mann-Whitney u-tests (grid > cloud) of all variables for test 2  223
5.9 Summary of experimental conditions for test 3  228
5.10 Unpaired Student t-tests (grid > cloud) of all variables for test 3  230
5.11 Summary of experimental conditions for test 4  236
5.12 Summary of experimental conditions for all audio experiments  245
## List of Icons

<table>
<thead>
<tr>
<th>License</th>
<th>Attribution</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Creative Commons] 3.0</td>
<td>WPZOOM Developer Icon Set</td>
<td><a href="http://www.wpzoom.com">http://www.wpzoom.com</a></td>
</tr>
<tr>
<td>[Attribution] 3.0</td>
<td>Icons4Android</td>
<td><a href="http://icons4android.com">http://icons4android.com</a></td>
</tr>
<tr>
<td>[Creative Commons] 3.0</td>
<td>WPZOOM Developer Icon Set</td>
<td><a href="http://www.wpzoom.com">http://www.wpzoom.com</a></td>
</tr>
<tr>
<td>[Creative Commons] 3.0</td>
<td>WPZOOM Developer Icon Set</td>
<td><a href="http://www.wpzoom.com">http://www.wpzoom.com</a></td>
</tr>
<tr>
<td>[Creative Commons] 3.0</td>
<td>Stephen Hutchings</td>
<td><a href="http://typicons.com">http://typicons.com</a></td>
</tr>
<tr>
<td>[Creative Commons] 3.0</td>
<td>Icons8.com</td>
<td><a href="http://icons8.com/android-icons">http://icons8.com/android-icons</a></td>
</tr>
<tr>
<td>[Creative Commons] 3.0</td>
<td>MIT License Open Iconic</td>
<td><a href="http://useiconic.com/open">http://useiconic.com/open</a></td>
</tr>
<tr>
<td>[Creative Commons] 3.0</td>
<td>Stephen Hutchings</td>
<td><a href="http://typicons.com">http://typicons.com</a></td>
</tr>
<tr>
<td>[Attribution] 3.0</td>
<td>WPZOOM Developer Icon Set</td>
<td><a href="http://www.wpzoom.com">http://www.wpzoom.com</a></td>
</tr>
<tr>
<td>[Attribution] 3.0</td>
<td>WPZOOM Developer Icon Set</td>
<td><a href="http://www.wpzoom.com">http://www.wpzoom.com</a></td>
</tr>
<tr>
<td>[Attribution] 3.0</td>
<td>Icons4Android</td>
<td><a href="http://icons4android.com">http://icons4android.com</a></td>
</tr>
<tr>
<td>[Attribution] 3.0</td>
<td>WPZOOM Developer Icon Set</td>
<td><a href="http://www.wpzoom.com">http://www.wpzoom.com</a></td>
</tr>
</tbody>
</table>

Legend: Ⓓ (Creative Commons) ⒱ (Attribution) ⒰ (Share Alike) Ⓥ (Non Commercial)
1 Introduction: context, terminology, concerns, statement

IF THIS WORD MUSIC IS SACRED AND RESERVED FOR EIGHTEENTH AND NINETEENTH-CENTURY INSTRUMENTS, WE CAN SUBSTITUTE A MORE MEANINGFUL TERM: ORGANIZATION OF SOUND.

JOHN CAGE ¹

Creativity sources its roots from inspiration. One of the oldest form of recording that transmits literally the thoughts of creators is written material, such as books, essays, manifestos, scientific publications, this thesis. But the creations themselves go beyond words, speak for themselves, visually, aurally. Recent works, from the previous and current centuries, have been recorded with techniques enabled by increasingly affordable signal processing technologies (cameras and microphones), or literally transmitted in realtime through complementary technologies (visual and auditory displays), live, as installations, performances. With such a flow of audiovisual content, how can we keep the pace and remain inspired?

In section 1.1 we will briefly go through conventions we chose to rule this manuscript. In section 1.2 we will explain the context surrounding this thesis, particularly the research fields contributing to its identity. In section 1.3 we will define terms grounding this thesis, most of which are present in the thesis title. In section 1.4 we will list the pending research concerns of browsing multimedia collections. In section 1.5 we will narrow down the scope of this thesis, starting with its statement. In section 1.6 we will summarize the contributions of this work.

1.1 Conventions

Here we explain some conventions we decided to follow through this whole manuscript.

1.1.1 Identities

This work, even if presented by a single person as a thesis, wouldn’t exist without contributions from other people. Throughout this document, the generic third person “we” will drive the sentences. Occurrences of “I” will underline choices made by the main author.

1.1.2 Layout

This manuscript uses the \texttt{tufte-latex} \LaTeX{} package based closely on Edward Tufte’s typesetting recommendations, but forked freely with adaptations. Our goal is to ease the task of the reader. One major improvement over usual scientific writings are references directly available in a margin, suppressing incessant back-and-forth navigation to the end of the document. Sentences end before pages. Layout has been optimized for double-page display.

1.1.3 Language, spelling

US English is employed, while visiting frequently my cousins in UK highly contributed to my will to improve my knowledge of the English language.

1.1.4 Typesetting

Project names that are artistic (movies, art installations...) or computer-related (hardware, software, tools, applications...) are mentioned in italics.

1.1.5 Referencing

The web is not pervasive: some websites have a “time to live” of a few years, with domain name changes and content management system updates. Thus we will avoid hyperlinks. Using keywords on web search engines will be faster than typing a full link anyway.

1.1.6 Avoid printing

I recommend the reader, “digital native” or not, to read this material on a computerized device for two reasons: it saves paper, and it improves the browsing experience (searching by keywords, zooming...), a concept shared with this thesis.
1.2 Context

Figure 1.1 shows the many disciplines related to interaction design and how they overlap each other. It is a slight adaptation of a poster made by design company Envis Precisely, itself inspired by Dan Saffer’s original drawing featured in one of his books. It greatly illustrates the many fields that can both contribute to or benefit from the research interests of this thesis.

The fields that can benefit from our research outcomes in the context of browsing media collections are creative practices that mediate visual design by (re)using media content. Sound design is an example. Sound designers need to organize their sound libraries, whether they authored them or sourced them from third-parties. Another example is video broadcast. Video editors need to edit long video rushes, the first time-consuming phase of their workflow is browsing the videos to find passages of interest. Complementarily, a book by Bill Moggridge analyzes how online services are designed to provide media content online, for less expert users.

The fields that can contribute to our research are numerous. Multimedia information retrieval, a subbranch of signal processing inheriting itself from electrical engineering, and also related to machine learning and data mining, supports our case by the study of how media elements interrelate based on signal properties. Data/Information visualization is a discipline attached to computer science and computer graphics, but also sourcing roots from cognitive science, focuses on how to present information to the user’s perception. Visualization can be considered as one means of interaction, one modality considered among others in the field of human-computer interaction.

Chapter 3 focusing on the method that we followed throughout this thesis, we will further explain into detail how findings in these fields can aid us to design more suitable media content browsers.
Figure 1.1: The Disciplines of User Experience Design, forked from Envis Precisely, itself inspired by Dan Saffer.
1.3 Terminology

Here follows an inductive approach to outline visually the key terms of this thesis. From all the references collected for this thesis, first a subset of references containing the term “browser” is filtered using the text search of a standard file browser. This subset of circa 200 items is then imported in the Zotero opensource bibliography manager. Using the Paper Machines extension by Chris Johnson Roberson can be generated a word cloud filtered by Term Frequency-Inverse Document Frequency (tf*idf) in Figure 1.2.

This visualization reveals the most frequently represented concepts. Here we list some of them by category then decreasing number of occurrences (from the font weight):

- content: music, video, audio, sound, media, image, multimedia, collection...
- user interface design: visualization, interaction, interactive, evaluation, participants...
- information retrieval: retrieval, similarity, distance...
- task; search, query, browsing, navigation...

Now from the thesis title, here are the terms that we are going to explain in further detail throughout this section:

- media (section 1.3.1),
- organizing content (section 1.3.2),
- similarity (section 1.3.3),
- navigation, browsing (section 1.3.4).
1.3.1 Media

“Multimedia data” commonly refers to content (audio, images, video, text...) recorded by sensors and manipulated by all sorts of end-users. In contrast, the term “multimodal data” describes signals that act as ways of communication between humans and machines. Multimodal data can be considered as of a subset of “multimedia data”, since the first are produced by human beings. Multimedia data thus broaches a wider range of content (natural phenomena, objects, etc...).

1.3.2 Organizing content

The founder of Mundaneum, Paul Otlet (1868-1944), among other notable achievements, wrote two treaties on how to classify knowledge, Traité de Documentation (1934) and Monde: Essai d’universalisme (1935), long before nowadays online search engines. He theorized a whole workflow on documentation, including: the Universal Decimal Classification, and 3×5-inch cards to label items in catalogs.

What could be written on such cards? Here follow examples of criteria used for the organization of media data:

- **metadata** usually provides factual data on media items, which can be generic such as: author, title, album or collection, location or geographical origin... and often the date of creation or publication, or specific to the media collection;
- **semantic** data, such as tags, add subjectivity to media elements, and can often be organized into **ontologies**, which provide a relational structure to data that can’t be classified into mutually-exclusive categories, for instance: musical instruments;
- by means of computer analysis and signal processing, **content-based criteria** are extracted from the data and stand as objective numeric data; from low-level criteria close to the signal properties as evocative as the algorithm designed to output these can be, to higher level including perceptual criteria adapted to the human perception: mean color or shape for images, motion orientation for videos, energy or loudness in audio...
1.3.3 Similarity

1.3.3.1 The concept of similarity through (digital) arts

Literally, "similarity" names one of the Gestalt Laws in cognitive psychology: the human visual perception tends to discriminate outsider elements from groups in a visual collection. It has high implications in human-computer interaction and objector graphic design, such as how people can remember the content of a scene by gaining a structured knowledge of it, and how people can be attracted by objects or visuals that remind them of features of other objects they might have liked previously.

More specifically, humans have ever been fascinated by the complexity of natural phenomena showing repeating patterns: lightning bolts, clouds, tree-shaped vegetables, viscous flows. This concept of "self-similarity" has been inspiring fractal art, and is salient in paintings from Escher and compositions from Bach, as examined by Hofstadter.\(^6\)

Each of both definitions, similarity as Gestalt law and self-similarity, underline one major aspect of similarity in media content: similarity can be used to characterize and compare several elements of a collection (inter-media), and focus on the structure and contents of one single element (intra-media).

Several aspects of similarity can be used to describe the nature of artworks. First the relation of art to people: who originated it versus who keeps it alive in people’s memories (the issues of authoring, composition, interpretation, performance, appropriation, inspiration, creativity, emulation, reproduction, recomposition, curation, restoration, preservation); second the relation of art works between themselves: what the work represents and how it place itself in a context of other works (identity, authenticity, singularity, resemblance). These concepts emerge in movie remakes (for instance re-performed or “sueded” movies in Michel Gondry’s Be Kind Rewind), montages (Orson Welles’ F is for Fake), collages (works of Jennifer and Kevin McCoy, Vicky Bennett aka People Like Us), cover bands and song covers (from re-interpretation to resampling with John Oswald’s Plunderphonics).

1.3.3.2 Computational similarity

Computational similarity analysis consists in providing a machine interpretation of the salient characteristics of media objects, by applying feature extraction algorithms that downsize the data contained within the media into threads of specific information, and comparing the distribution of these features over a collection using adapted distance metrics. Features can be content-based, that is extracted directly from the digital representation of the media object; or semantic-based, provided by manual annotation and labels.
1.3.4 Navigation, browsing

Marchand-Maillet et al. provide a recent and detailed overview on the state-of-the-art of interactive representation of media databases, focusing on image browsing applications. It concludes mainly that most applications still miss a proper user-friendly interaction. They define browsing as a directed task aiming at finding a specific and intended target, while navigation refers to a more exploratory task aiming at discovering a collection of items. In her book on search user interfaces, Hearst discusses several theories and frameworks modeling search, and differentiates querying/searching versus browsing/navigating: the first behavior ends up in generating news collections or gatherings of information, the second covers getting information from existing groups.

More precise terms have been introduced to define several navigational approaches:

- **Scrolling** describes the linear navigation, often in one direction, in (web) pages, thus perceived as spatial control. Wilson recommends to take into consideration that “searchers rarely scroll”, so important information should be located above the first-scroll point.

- **Pan, zoom and rotation** are classic map-based navigation techniques, the first two are recurrent for browsers.

- **Skimming**, coined for speech navigation, initially employed for locating sounds events while listening to recorded tapes at low speeds, characterizes the navigation in temporal signals with frequent jumps in time, mostly positional control.

- **Skipping** consists in rejecting segments of no interest.

- **Cueing** is the process of identifying the boundaries of sections of interest.

---


1.4 Concerns of browsing media collections

An anticipated conclusion from our state-of-the-art chapter 2: there is no widely-accepted tool involving audio or video browsing that features in-depth content-based organization by similarity. On the one hand, Apple recently introduced face and scene detection in their video editing tool Final Cut X. On the other hand, Adobe and Princeton’s video tapestries summarization technique published in 2010 [12] is still not integrated in Adobe tools. Is it because these tapestries take multiples of the video duration to compute? Are there other reasons?

Some concerns that are not of direct interest for this thesis may partly answer this question.

• Content-based features need to be more robust, more high-level.
• The computational time associated to the content-based workflow needs to be reduced.
• The scalability to massive “big data” datasets needs to be considered, with accessibility of indexing results through cloud-based services.

Complementarily to these concerns, here follow these that we feel are missing as well, but received less attention or are of concern for smaller research communities (interactive or performative media research), and are directly of our interest.

• Techniques for content-based organization that have been appearing in the multimedia information retrieval community need to be evaluated with users using a methodology from the human-computer interaction community.
• Design cues from information visualization should be borrowed to improve representations of collections that often directly plot the results of dimension reduction over the feature set extracted from media collections.
• Browsing and navigation should be made not only efficient, but also attractive to the user. This may be attained by designing not only how information is transmitted visually, but also manipulated gesturally.

---

1.5 Thesis: statement, scope

1.5.1 Statement

This thesis aims at creating browsers for media collections organized by content-based similarity that end up being useful for media practitioners, particularly for sound designers. We believe that such tools can be better designed by inter-disciplinarily combining techniques from the fields of multimedia information retrieval and human-computer interaction (including information visualization and gestural interaction). To prove our point, we propose to evaluate through user experiments with experts a subset of the tools we have been prototyping iteratively, relying on MediaCycle, a framework emerging from such projects.

1.5.2 Scope

1.5.2.1 Organization: content-based rather than semantic

Media collections are often heavily tagged by expert users themselves so that they can find relevant content more easily. If these annotations are properly managed, carefully chosen from taxonomies or ontologies, we believe that filtering a collection by tags is an ideal way of starting a search session, until search results need to be reviewed. We focus on studying this reviewing tasks, that can be supported by content-based analysis can, complementarily to the semantic approach.

1.5.2.2 Audio and video as chosen media types

In his PhD thesis, Filatriau argues that instructive parallels may be drawn between image and sound processing fields. Goodrum also reviews some lessons from video retrieval evaluation that could be extended to music information retrieval. Audio and video content being temporal by nature, we believe that similar navigation techniques may benefit to both types.

1.5.2.3 Visual display and gestural interaction as chosen modalities

In his book, Moggridge mentions that "the VCR is the most frequently mentioned scarecrow of interaction design difficulty - so few people get past the play function and actually record successfully", due to a lack of physical metaphors representing its available controls. We believe that visual display and gestural interaction can be designed to be paired into user interfaces that can well support media content browsing.

1.6 Contributions

Throughout this thesis, we produced the following contributions:

• scientific:
  – a comprehensive state-of-the-art on media browsers (in chapter 2);
  – user evaluations (described in section 5.2): 1 pilot evaluation with an intra-media video browser part of a live competition and 4 iterative evaluations with an inter-media audio browser;
  – publications (listed in section 1.6.1): 1 journal paper as co-author, 13 international conference papers as first author, 5 international conference papers as co-author;

• creative and artistic: work with 5 experts concerning artistic and creative applications of our work, including contextual inquiries and qualitative evaluation of our research efforts (described in section 5.1);

• technological: significant development of the MediaCycle framework for content-based media browsing in collaboration with many lab colleagues, including 9 projects implementing browsers for audio or video content (described in chapter 4);

1.6.1 Publications

1.6.1.1 Journal paper of an international collaboration

We collaborated with international researchers from the field of multimedia information retrieval to write a journal article on a competitive evaluation of video browsers:


1.6.1.2 Conference papers

The following lists of publications are sorted first by decreasing year, then by decreasing h5-index and h5-median retrieved from Google Scholar in February 2015 (whenever available). Annual acceptance rates are mentioned whenever available. Quoting the Google Scholar website: “h5-index is the h-index for articles published in the last 5 complete years. It is the largest number h such that h articles published in 2009-2013 have at least h citations each. h5-median for a publication is the median number of citations for the articles that make up its h5-index.”

http://scholar.google.com
1.6.1.2.1 As first author

The author of this thesis lead the co-authorship of scientific publications and presented these to national and international conferences:

  Google Scholar 2015 h5-index (31) / h5-median (43) Acceptance rate 2014 (overall): 47.7%

  Google Scholar 2015 h5-index (24) / h5-median (31) Acceptance rate 2014 (overall): 33%

  Acceptance rate 2014 (overall): 59%

  Google Scholar 2015 h5-index (14) / h5-median (18)


• Christian Frisson, Stéphane Dupont, Xavier Siebert, Damien Tardieu, Thierry Dutoit, and Benoit Macq. “DeviceCycle: rapid and reusable prototyping of gestural interfaces, applied to audio browsing by similarity”. In: *Proceedings of the New Interfaces for Musical Expression++ (NIME++)*. Sydney, Australia, 2010

1.6.1.2.2  As co-author

The author of this thesis participated to the writing of co-authored papers presented to national and international conferences:

• Stéphane Dupont, Thierry Ravet, Cécile Picard-Limpens, and Christian Frisson. “Non-linear dimensionality reduction approaches applied to music and textural sounds”. In: *IEEE International Conference on Multimedia and Expo (ICME)*. San Jose, USA: IEEE, 2013. doi: [10.1109/ICME.2013.6607550](https://doi.org/10.1109/ICME.2013.6607550)  
  2015 h-index (23) / h5-median (28)  
  Acceptance rate 2013 (overall): 30%

  Google Scholar 2015 h-index (23) / h5-median (28)  
  Acceptance rate 2009 (poster): 33%

  Google Scholar 2015 h-index (11) / h5-median (17)


1.6.1.3 Doctoral consortiums

The author of this thesis presented his doctoral work to the following doctoral consortiums part of national and international conferences:


1.6.1.4 Workshop papers

1.6.1.4.1 As first author

The author of this thesis lead the co-authorship of scientific publications and presented these to national and international workshops:


• Christian Frisson, Sema Alaçam, Emirhan Coşkun, Dominik Ertl, Ceren Kayalar, Lionel Lawson, Florian Lingenfelser, and Johannes Wagner. “CoMediAnnotate: towards more usable multimedia content annotation by adapting the user interface”. In: Proceedings of the eNTERFACE’10 Summer Workshop on Multimodal Interfaces. Amsterdam, Netherlands, 2010

1.6.1.4.2 As co-author

The author of this thesis participated to the writing of co-authored papers presented to national and international workshops:

1.6.1.5 Quarterly progress scientific reports (author and copyeditor)

The author of this thesis participated in 3-month numediart projects and the co-authoring of their reports, and copy-editing most of the quarterly progress scientific reports including other trimestrial project reports:


• Christian Frisson, Onur Babacan, and Thierry Dutoit. “MakamCycle: improving the understanding of Turkish Makam music through the MediaCycle framework”. In: *QPSR of the numediart research program*. Ed. by Thierry Dutoit. Vol. 5. 2. June 2012, pp. 17–20
1.7 Conclusion

In this introductory chapter, we positioned this thesis in its context, explained some key terms that are recurrent in related works, clarified the concerns in browsing media collections, delimited the scope of our work by the issues it aims at solving, and summarized our contributions.

In chapter 2, we will investigate more into detail over research works that constitute the background of our own work: an overview on systems for browsing media content.

In chapter 3, we will explain the method we chose to assist us in solving our research questions: combining interaction techniques (gestural input and visualization) with a content-based organization dataflow.

In chapter 4, we will describe our prototyping environment and the various media browsers it enabled.

In chapter 5, we will analyze our experimental results from the evaluation of some of our media browsers with users.

In chapter 6, we will conclude on our findings and provide clues for future works.
2 Background: browsing media content (by similarity)

In a historical loop, the computer has returned to its origins. No longer just an analytical engine, suitable only for crunching numbers, it has become Jacquard’s loom – a media synthesizer and manipulator.

Lev Manovich

In the current chapter, we will compare existing systems that aid users to browse media content, mainly audio and video collections, not necessarily relying on content-based similarity, but focusing on it.

So as to restrict the scope of the current chapter, we will focus only on fully-fledged standalone systems.

The next chapter will complement this approach: we will theorize more on a subset of modalities, that is visualization (section 3.2) and gestural input and control (section 3.3). This will set a basis for deeper analysis of such applications, inductively, while the following analysis is deductive. We believe that improving the user interfaces specifically on these modalities can drastically improve the user experience of media content applications.
2.1 A taxonomy of media browsers

There are many criteria at hand to present media browsers and hopefully illustrate and discover patterns. We selected several categories to define a taxonomy. These can be summarized in a tight visual space as in table 2.1. A legend of entries for each category is illustrated in table 2.2.

<table>
<thead>
<tr>
<th>media</th>
<th>sound (stimulating audition)</th>
</tr>
</thead>
<tbody>
<tr>
<td>media</td>
<td>music (stimulating audition and the musical mind)</td>
</tr>
<tr>
<td>media</td>
<td>video (stimulating vision spatiotemporally)</td>
</tr>
<tr>
<td>media</td>
<td>image (stimulating vision spatially)</td>
</tr>
<tr>
<td>media</td>
<td>text (stimulating vision and the lingual mind)</td>
</tr>
<tr>
<td>granularity</td>
<td>inter-media (between files or fragments)</td>
</tr>
<tr>
<td>granularity</td>
<td>intra-media (inside a file or a fragment)</td>
</tr>
<tr>
<td>organization</td>
<td>content-based (by signal processing)</td>
</tr>
<tr>
<td>organization</td>
<td>context-based (from metadata)</td>
</tr>
<tr>
<td>visualization</td>
<td>represented through ( n ) dimension(s) (from 0D to 2D)</td>
</tr>
<tr>
<td>interaction</td>
<td>Windows Icon Menu Pointer (WIMP), currently as in everyday desktop applications</td>
</tr>
<tr>
<td>interaction</td>
<td>Zoomable User Interfaces (ZUI), as in digital map applications</td>
</tr>
<tr>
<td>interaction</td>
<td>Tangible, Embedded/Embodied Interaction (TEI), with controllers, beyond the desk</td>
</tr>
<tr>
<td>users</td>
<td>novices</td>
</tr>
<tr>
<td>users</td>
<td>experts in creative practices: sound designers, video editors...</td>
</tr>
<tr>
<td>setting</td>
<td>desk, room, venue</td>
</tr>
<tr>
<td>community</td>
<td>multimedia/music information retrieval (MIR) and digital signal processing (DSP)</td>
</tr>
<tr>
<td>community</td>
<td>human-computer interaction (HCI) and information visualization</td>
</tr>
<tr>
<td>community</td>
<td>companies and the industry</td>
</tr>
<tr>
<td>community</td>
<td>artistic works</td>
</tr>
<tr>
<td>availability</td>
<td>price or software license</td>
</tr>
<tr>
<td>usability</td>
<td>qualitative evaluations: through surveys, questionnaires...</td>
</tr>
</tbody>
</table>
| usability | quantitative evaluations: through logged data (time, task success...)

Table 2.1: Media browser taxonomy template table

The List of Icons (part of the front matter of this thesis) details the licenses under which each icon (above and thereafter) is released.
2.2 An overview of media browsers

The goal of this overview of media browsers is to grasp the complexity of our problem and to get inspired by other designs while prototyping ours. Our selection of reference media browsers will be presented first categorized in the general trends they belong to (by type of organization, by media type), then chronologically. For the readers’ convenience, one browser will be presented per page, beginning with a picture or screenshot of the system to help the reader understand the general design, assorted with a margin table summarizing its taxonomy following the template explained just before, both positioned repeatably on top of pages. This layout facilitates rapid scanning.

After an introduction on why we need to organize files illustrated by the metaphor of the physical desktop (section 2.2.1), we will start with context-based browsers, for files (section 2.2.2), video (section 2.2.3) and audio (section 2.2.4). We will proceed with an intermission on some tools for digital libraries (section 2.2.5). We will continue with content-based browsers for audio (section 2.2.6) and video (section 2.2.7). We will finish with freehand tools (section 2.2.8) and content-based artistic works (section 2.2.9).

2.2.1 The metaphor of the desktop

Before the advent of computers and digitized content, people would sort their crops in buckets or piles, arrange their spare parts and collectibles in shelves or boxes, line up books in libraries. In settings such as offices, desktops, creative spaces; in times of getting things done, people still sort physical documents in piles over the desk or on the ground, so as to classify these, or decide upon actions related to them, in a “do/delegate/defer/delete it” modus operandi.  


2.2.2 Context-based file browsers

File browsers across the many operating systems would follow this desktop metaphor, providing views of folder containing files. Moggridge retraced the history of the design of some file browsers. Some designs went beyond the file and folder “iconification” on a flat space reminiscent of documents over a table by borrowing more cues of interaction with documents. Context-based denotes all information that is added to media content beyond file properties: metadata, tags, hyperlinks... We will thus proceed with investigating media browsers that use metadata- and user-defined organization.

2.2.2.1 WebBooks in the Web Forager by Card et al. (1996)

WebBook 4 by Card et al. was the first 3D interactive visualization of a page-based document slightly above the page level: when flipping pages. Their Web Forager sorted documents on a 3D hierarchical workspace, combining into a single view the main computer desktop view and their application switching mode, that are common nowadays in desktop environments.

In their book on science-fictional interfaces 5, Shedroff and Noessel analyze how science fiction movies depict interaction design, particularly for our interest: file browsers. Among these, one mainstream movie, Jurassic Park by Steven Spielberg, featured a 3D browser borrowed from UNIX systems.

2.2.2.2 Raskin file browser (2001)

Jef Raskin (1943-2005), a usability specialist and professor at UCSD, notably established recommendations to make interfaces more *humane* \(^6\) and notably invented the drag and drop technique at Apple.

A Swiss company recently got inspired by his works and named their product after him, *Raskin* \(^7\). *Raskin* follows up on the desktop metaphor and allows to sort files visually and freehand, by user-defined clusters.


\(^7\) [http://raskinformac.com](http://raskinformac.com)
2.2.2.3  

BumpTop 3D file browser by Anand Agarawala (2006)

Agarawala’s Master Thesis 8 provides some insight on works that got inspired by the metaphor of the desktop to propose tangible methods to organize data in piles. One output of his work is BumpTop, a 3D file browser resembling a tabletop or room, following the concept of piles of documents. BumpTop got acquired by Google in 2010, its development stalled and was transferred to the open source community 9.

8 Anand Agarawala. “Enriching the Desktop Metaphor with Physics, Piles and the Pen”. MA thesis. Graduate Department of Computer Science, University of Toronto, 2006

9 https://code.google.com/p/bumptop/
2.2.2.4 Liquifile by Waldeck et al. from Liquiverse/iVerse (2004)

Liquifile\textsuperscript{10}, from Waldeck et al.\textsuperscript{11} who later founded German company Liquiverse/iVerse, allows to explore documents through other dimensions: in addition to the two-dimensional tree and table views frequent on standard file browsers, an extra pane on the side of these view sorts documents on a timeline, time in the horizontal axis, the position on the tree or table view in the vertical axis. This browser was initially tailored for pen-based devices, particularly mobile devices earlier to smartphones such as PDAs, convenient for these small form factors preceding the currently widespread multitouch surfaces. These browsers offer the advantage of condensing the information space. Node opacity and overlap can be uncluttered at mouse or pen hover. To do so, instead of magnifying the size of nodes using a lens technique, their “liquid browsing” technique adapts the distance between nodes by repulsing these from the pointer.

\textsuperscript{10} \url{http://www.liquifile.info}

2.2.3 Context-based video browsers

2.2.3.1 Film Finder by Ahlberg and Shneiderman (1993)

Ahlberg and Shneiderman’s Film Finder is probably one of the oldest context-based media browsers. Based on metadata, it allows to sort movies on a Starfield Display, a traditional scatter plot augmented with zooming and selection capabilities. Film Finder led to a commercial product named SpotFire, driven by Ahlberg, as explained in a report of a debate between Ben Shneiderman putting forward direct manipulation and Pattie Maes advocating interface agents. A quote from Shneiderman: “I think we would do best to focus on the remarkable human capabilities in the visual domain, which I think are largely underutilized by the current designs with 40 icons in 2-3 windows. I think we should have two or three orders of magnitude more: 4,000 or more items on the screen in an orderly way that enables people to see all of the possibilities and navigate among them.” Shneiderman’s works on recommendations on information visualization influenced lots of subsequent projects, for instance after his suggestion: “overview first, zoom and filter, then details on demand”.

13 http://www.spotfire.com
2.2.3.2 AAU Video Browser by Del Fabro et al. (2010)

The AAU Video Browser by Del Fabro et al. aims at questioning the necessity of time-consuming content analysis for search tasks by providing a simple interface without content-based analysis. It offers two views: the Parallel View visualizes simultaneously as many video segments of equal length and evenly distributed in the original source video (or video collection) that can be positioned on a grid, useful for getting an overview of a long video or video collection. Selecting one segment takes the hierarchy of segmentation to the next level: the grid view plays sub-segments of this segment. The Tree View gives an insight on the levels of segmentation hierarchy at one glance by assigning rows to sub-hierarchies. Starting with one row, clicking on one video segment adds another row refining the chosen segment.
2.2.3.3 Panopticon by Jackson et al. (2013)

Jackson et al.\textsuperscript{16} recently compared three interactive visualizations of single video files: 1) as consumer-grade baseline the current YouTube video player running on any web browser, which overlays thumbnails corresponding to positions on its seek bar (when hovering the bar or moving the cursor); 2) as research-grade baseline, the VideoBoard which orders in a grid static thumbnails of the video evenly-sampled, with on demand playback, associated with a standard video player; and 3) their design: the Panopticon which complements the latter by making the thumbnails dynamic, looping in sync over their equal time span. An evaluation was conducted with 36 participants each assigned to one design, asked to find specific verbally-described passages of videos (1 feature film, 1 surveillance video, 1 unedited footage), measuring search time and eye movements. These 3 designs were ranked in terms of decreased mean search times (Panopticon scored better than VideoBoard itself better than the baseline) for the 3 video types (significantly for surveillance videos, less for unedited footage, not for narrative movies). They improved their system by replacing the associated video player by an overview mode directly on the grid view and increasing the size of the thumbnails.

2.2.4 Context-based sound browsers

2.2.4.1 SonicBrowser by Fernström and Brazil (2001)

Fernström and Brazil introduced the Sonic Browser \(^{17}\) focused on information visualization. A 2D starfield display allows to map the metadata of audio files to visual variables. Its Hyper-Tree view consists in a spring-layout hierarchical graph visualization for browsing the file tree of sound collections. One of its particular features is that multiple streams of audio can be played simultaneously, depending on the user-defined diameter of the aura, a circular region assigned to the pointer.

They qualitatively evaluated these views with 15 students through timed tasks and a questionnaire \(^{18}\); and compared their system against the Microsoft Windows 2000 explorer through a think-aloud protocol with 6 students (commenting their actions and appreciations while testing). We couldn’t trace reports of evaluations of the specific quantitate contribution of content-based organization. They however showed through a user evaluation that multiple stream audio feedback significantly outperformed single-stream audio feedback \(^{19}\).

They later approached content-based organization through the Marsyas framework (see Section 2.2.6).
2.2.4.2 Visualizations of music libraries by Torrens et al. (2004)

Torrens et al. proposed three interactive visualization techniques to assist users in browsing their music collections, relying solely upon metadata (genre, artist, track, duration). The *Disc Visualization* lays out tracks in a polar view, the angle corresponding to the genre and the radius to the year of release. The *Rectangle Visualization* uses a similar mapping but with cartesian coordinates. The *Tree-Map Visualization* recursively splits the area into sub-rectangles, from genres to sub-genres and artists. In the paper, the authors discuss the advantages and drawbacks of each visualization, such as the density of the layout and practicality of zooming. User studies are claimed to be planned as future work.

The authors incorporated into a company named Strands filed a patent in 2006 that got accepted in 2010, restricting the usage of the 3 views in the context of music libraries.

---


21 [http://strands.com](http://strands.com)

In 23, Stewart et al. propose a 3D browser of audio content without visualization, solely based on auditory display. Sounds are triggered by people walking in a room. This first version was created with the Max/MSP modular framework and controlled with a Nintendo Wii remote controller. Sounds are spatialized and an *aura*, or selection circle of variable width (adapted using buttons of the remote control), renders the sounds within its periphery. The remote control vibrates whenever pointing to a sound source (its direction guessed from accelerometers) and provides stereo playback. One of the conclusions they obtained through an evaluation with 12 users is that the mapping of the free-form gestures with the remote control to the virtual audio space wasn’t ideal.

Later, the authors refined their design, renamed it the *amblr* 24 and patented it 25. They moved to a client-server architecture with a minimalistic graphical user interface built with the Processing Development Environment. Housing the remote controller in an off-the-shelf steering wheel enclosure (designed for car video games) suggested how the it should be held and improved the gestural interaction over the previous design.
2.2.4.4 AudioFinder by Iced Audio (1986)

AudioFinder by Iced Audio\(^{26}\) mimics personal music managers such as Apple iTunes: on top a textual search input widget allows to perform a query, a top pane proposes a hierarchical view similar to the “column” view of the Finder to browse the file tree of the collection, a central view features a spreadsheet to order the results along audio and basic file metadata, a left pane lists saved results like playlists. A bottom row offers waveform visualizations and the possibility to apply audio effect processing to quickly prove the potential variability of the sounds before dropping these into other applications such as digital audio workstations.

\(^{26}\) [http://www.icedaudio.com](http://www.icedaudio.com)
A major product that is of widespread use in the creative industries is Soundminer HD\textsuperscript{27}. It provides a similar interface to AudioFinder (see \ref{sec:audiofinder}), plus an alternative layout named 3D LaunchPad, illustrated in the screenshot above, that allows similarly to the Apple Finder CoverFlow view to browse sounds (instead of songs) by collection (instead of album) cover, with the difference that the former is a 2D grid and the latter a 1D rapid serial visualization technique.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{soundminer_launchpad.png}
\caption{Screenshot of Soundminer HD 3D LaunchPad layout.}
\end{figure}

\textsuperscript{27} \url{http://www.soundminer.com}
Adobe is yet another company facilitating creativity. With Bridge, it provides a more general digital asset management solution that are accessible through their entire application suite. It allows advanced text-based filtering queries. Bonus integrated features are a batch conversion engine, a metadata management tool, a sound design tool with DSP processing capabilities, server and web publishing functionalities. It focuses on production-required capabilities and avoids content-based functionalities.

It features several layouts. The Essentials layout sorts media thumbnails in a grid ordered by metadata (filename, date, size... and color profiles for visual media). The Filmstrip and Preview (pictured above) layouts allow rapid serial visual presentation similarly to the Apple Finder CoverFlow: media elements are ordered in a row (or a column) and instantly previewed on a larger pane when selected. The Metadata and Keyword layouts associate a spreadsheet view with files against their metadata to a tree view of respectively metadata or keywords (with possible hierarchies).

---

28 [https://creative.adobe.com/products/bridge]
2.2.4.7 Vintage Radio Interface by Hopmann et al. (2012)

With their *Vintage Radio Interface* \(^29\), Hopmann et al. simplified the browsing experience to its minimalistic essence: a one-dimensional controller, a Griffin PowerMate device, allows to browse a list of music tracks ordered hierarchically by genre, artist, album, track, as in conventional digital music managers, but without panels. Its design was inspired by old analog radios, from both hardware and software aspects: a small screen embedded in a wooden enclosure, a timeline of albums resembling the tuner display reusing frequency graduations to mark tracks and albums. Audio static is used as feedback when “tuning” to desired songs. Through a qualitative evaluation with 24 users, they concluded that organizing songs first by genre can be confusing for targeted search, but is useful in exploratory mode not to break moods induced by genres. Testers expressed wills to associate the knob-based control to a touchscreen (to fine-tune more browsing parameters) and/or a remote controller. This browser sets itself apart from others by opting to browse media content in 1D.

---

2.2.5 Tools for digital media libraries

Lesk et al. analyze in 30 the foundations of digital libraries, with chapters developing on: their evolution, their diverse media types (from text to images), multimedia storage and retrieval constraints, knowledge representation schemes, usability and retrieval evaluation, user needs, preservation, intellectual property, and future aspects such as creativity. We are concerned by a subset: organization, representation, evaluation.

2.2.5.1 Greenstone by Witten et al. (2000–)

Witten et al, in 31, members of the New Zealand Digital Library Project, describe how Greenstone 32 can be an open source solution to support digital libraries, combining a server application and reader/librarian user interfaces. Since it has been released as open source project in 2000, it has been featured in many worldwide projects, in cooperation with UNESCO and the Human Info NGO in Belgium. Its browsing user interface resembles basic file browsers with tree views. Some of its authors have very recently been integrating music information retrieval and linked data capabilities into Greenstone 33.

---


32 http://greenstone.org

2.2.5.2 Bohemian Bookshelf by Thudt et al. (2012)

The Bohemian Bookshelf by Thudt et al. encourages “serendipitous” findings, unexpected results while exploring collections of books in a library setting. To achieve so it proposes five visualizations. The Cover Colour Circle orders the book covers represented by a dot of their main colour, progressively uncovered from an intermediate circular preview to the full cover. The Keyword Chains visualizations represents the relationships between books in a subset of the collections based on their keywords. The Timelines visualization connects the publication time to the content time for each book. The Book Pile orders books by increasing page count, from bottom to top, and alternating left and right. The Author Spiral organizes books alphabetically by author, visually similar to a parchment role, with swirled ends.

They installed Bohemian Bookshelf in a library on a touch display. They interviewed 11 visitors through a questionnaire and recorded the time spent interacting per visitor from the 94 people who tried among 129 who approached the device. A frequent request from visitors is the will to switch between open-ended “serendipitous” exploration and targeted search. A concern is how the number of simultaneous visualizations and the complexity of the user experience are related.

2.2.6  Content-based audio browsers

Two research communities gather works on content-based media browsing: “music information retrieval” and “multimedia information retrieval”, focusing respectively on audio and video content. We will notice throughout this section that similarities can be drawn.

2.2.6.1  SpeechSkimmer by Barry Arons (1993)

One of the earliest content-based audio browsers must be SpeechSkimmer by Barry Arons. Its purpose is to get the gist of speech recordings without listening to these in their entirety. Using energy and zero-crossing rate as extracted features, speech intervals could be detected from ambient noise (assuming clean speech-only recordings). Speech segmentation was based on the pause durations (between words and sentences) and pitch variation (usually peaking when starting new topics). Four levels of skimming were offered: 1) original playback, 2) pauses shortened or removed (to/under a threshold of 500ms), 3) just a few seconds after each significant pause, 4) or after significant pitch variations. Tested with mice, trackballs, joysticks; the preferred input device was a touchpad mapped with vertical slices assigned to each of the aforementioned skimming levels (backward to forward), plus vertical fine-tuning inside each slice. Strips of paper were added as tactile guides to each level, making the user interface not dependent of the visual feedback, a primary design cue of this fish ear system. This design was evaluated by a think-aloud study with 12 people.
Another pioneering application is *SoundFisher* by Muscle Fish (1996), a start-up of scientists that graduated in the field of audio retrieval. Their application allowed to categorize sounds along several acoustic features (pitch, loudness, brightness, bandwidth, harmonicity) whose variations over time are estimated by average, variance and autocorrelation. Sounds are compared from the Euclidean distance over these features. The browser offers several views: a detail of sound attributes (filename, samplerate, file size...) in a spreadsheet, a file tree organized by folders or user-defined categories, and a scatter plot to sort sounds along one feature per axis. Queries can be textual (filtering by metadata or features) or by example (selecting one or more similar sound(s)).

---

37 [http://www.soundfisher.com](http://www.soundfisher.com)
38 [http://www.musclefish.com](http://www.musclefish.com)
2.2.6.3 MARSYAS3D by Tzanetakis et al. (2001)

MARSYAS3D\textsuperscript{39} is a browser for large collections of sounds that builds upon one of the oldest open source frameworks for music information retrieval: MARSYAS. Collections are displayed visually using a 8 by 18 foot rear projection screen providing a resolution of 4096 x 1536 pixels; and aurally through a 16-speaker surround system.

MARSYAS provides tools for “classification, segmentation, similarity-retrieval, thumbnailing, principal component analysis (PCA), beat-detection and clustering”. Multiple browser views are offered in the MARSYAS framework: a Standard list of filenames; a Tree generated by automatic classification; TimbreGrans that represent features in vertical color stripes concatenated horizontally and optionally overlaid on waveforms; TimbreSpace\textsuperscript{2D} and TimbreSpace\textsuperscript{3D} that represent each audio item respectively on 2D and 3D views with color, position and shapes mapped to audio features; and the SoundSpace that spatializes the sounds aurally.

\textsuperscript{39} George Tzanetakis and Perry Cook. “Marsyas3D: a prototype audio browser-editor using a large scale immersive visual and audio display”. In: Proc. Int. Conf. on Auditory Display (ICAD). 2001
2.2.6.4 SmartMusicKIOSK by Goto (2003)

SmartMusicKIOSK by Goto \(^{40}\) bears better its name when presented on a tablet PC. It was aimed at superseding the public listening stations in music libraries or shops to preview albums before buying or renting, that were limited with standard playback capabilities (fast-forward, previous/next song). By means of music analysis (Short-Time Fourier Transform power spectrum summed over 6 octaves to form Chroma features), it allows to segment songs into chorus and verse, that its graphical interface displays similarly to music sequencers, using an horizontal timeline.

The interactive display, actually pen-based (before the era of consumer-grade multitouch devices), was assorted with an external keypad seemingly duplicating the playback commands. The paper indicates the presence of an evaluation of the operation of the system, but doesn’t mention any number of subjects tested. Surprisingly, the next related work of the author (see 2.2.6.6) has been presented to the MIR community (here HCI) and underwent a qualitative usability evaluation.

CataRT 41 is an application by Diemo Schwarz developed in the Max/MSP modular dataflow framework, “mosaicing” sounds into small fragments for concatenative synthesis, that is reordering fragments following their content-based properties to create new synthesized content. Of the applications examined in this section, it is one of the most sustained in terms of development, and has been tested with many interaction techniques and devices 42. A 2D scatter plot allows to browse the sound fragments, with user-definable features assigned to the axes. The authors recently applied a distribution algorithm 43 that optimizes the spreading of the plotted sounds by means of iterative Delaunay triangulation and a mass-spring model, so as to solve the non-uniform density inherent to a scatter plot, and to open new perspectives for non-rectangular interfaces such as the circular reacTable and complex geometries of physical spaces to sonify. To our knowledge, no user study has yet been published for this tool. It is however claimed as future work in the aforementioned paper. It is available under a GPL license 44.

Figure 2.21: Screenshot

CataRT by Schwarz et al. (2004)


http://sourceforge.net/projects/ftm/

http://sourceforge.net/projects/ftm/
Musicream by Goto and Goto is an application for serendipitous discovery of songs. Following the metaphor of water (Musicream being a contraction of music and stream), songs fall like drops from 3 taps. The aperture of each three taps can be adapted, these are mapped to mood in the current implementation. Musicream relies on the MARSYAS framework for content-based analysis, chosen features are: MFCCs and spectral centroid/rolloff/flux/zero-crossings averaged along songs, durations of low-energy intervals, pitch periodicity and beat periodicity. These features, reduced to two polar coordinates from the two first components obtained through Principal Component Analysis (PCA), are mapped to the hue and saturation color circle (respectively angle and radius). It features 4 salient functionalities: music-disc streaming (songs streaming from the taps), similarity-based sticking (moving a given song attracts more easily similar songs), meta-playlist (advanced playlist management) and time-machine (recalling past listening activities). A qualitative evaluation was undertaken with 27 subjects.

---

2.2.6.7  MusicMiner by Moerchen et al. (2005)

MusicMiner \(^{46}\) relies on the Emergent Self-Organizing Map algorithm and uses two of its common representation methods, the U-Matrix (visible on the screenshot) and U-Map (looking more like the surface of a planet), to lay out songs on a map following the metaphor of a geographical map: valleys represent cluster of similar songs, mountains denote the boundaries between these clusters. Such visualizations are embedded in an application reminiscent of standard music library players and playlist managers. As content-based features, it relies on variants of MFCCs, on different representation scales than the Mel, such as Bark. It has been released as open source project \(^{47}\).

\(^{46}\) F. Moerchen, A. Ultsch, M. Noecker, and C. Stamm. “Databionic visualization of music collections according to perceptual distance”. In: Proc. of ISMIR. 2005

\(^{47}\) http://musicminer.sourceforge.net
2.2.6.8 Search Inside the Music by Lamere and Eck (2007)

Search Inside the Music, by Lamere (who later joined Echo Nest) and Eck (both at Sun Labs at the time), proposes several 3D interactive visualizations. The Album Cloud orders similar albums in a 3D view with position and distances conveying similarity (reduced to 3D using a multi-dimensional scaling technique). The Album Grid is a condensed version aligning these neighbors on a tighter grid so that these are adjacent under “overall/album stresses”. The Album Spiral is one example of mapping of the latter on a non-planar geometry, such as, in this case, a spiral.
MusicRainbow by Pampalk and Goto (2006)

MusicRainbow by Pampalk and Goto 49 proposes a simple user interface for browsing music songs: a colored disc visualization associated with a Griffin PowerMate rotary knob controller. The user interface is developed with the Processing Development Environment. The similarity between artists is computed from features extracted from their songs, notably loudness fluctuation and spectral shapes. First arranged on a 2D space, all artists are connected altogether by a shortest path whose contour is projected on a circular representation. Several colored disc sections correspond to tags associated to the artists, the color intensity mapped to the most frequent terms fetched using the Google search engine. By manipulating the controller, dialing allows to select an artist, pushing starts the playback. In this paper, user studies are planned as future work.
2.2.6.10  *Globe Of Music by Leitich and Topf (2007)*

Inspired by Geographic Information Systems, *Globe Of Music* \(^\text{50}\) by Leitich and Topf uses a spherical self-organizing map (SOM) to display songs on a globe. Regarding feature extraction, the *Statistical Spectrum Descriptor* (SSD) composed of seven statistical moments (mean, median, variance, skewness, kurtosis, min- and max-value) for the 24 critical bands is used. A qualitative user experiment was performed with 12 subjects, using a questionnaire and semi-structured interview. The users generally perceived the presence of a structure or organization in the positioning of the music songs, and underlined they could lead to memorizing items through the visual artwork associated to their node.

2.2.6.11 MusicBox by Anita Lillie (2008)

MusicBox is the audio browser developed by Anita Lillie throughout her Master thesis at MIT. By means of Principal Component Analysis (PCA), songs are represented in a 2D space, organized along several features fetched from the Echo Nest Analyze API (content-based) and Last.FM (context-based) such as: duration, number of sections, tempo, timbre, genre, rhythm histogram, moods, user-assigned tags, popularity. Node colors are tied to genres.

MusicBox differentiates itself from other 2D browsers by providing advanced selection tools, reminding of those from image manipulation application. The \textit{path-select} tool selects all songs encountered when drawing a line. Area selection tools fish all the songs contained in a rectangular or circular area. The \textit{create a tour} tool allows to discover the library by selecting a few tracks distant enough from each other. The \textit{wander from here} tool alleviates the randomness of the usual shuffle playlist with a user-defined maximum \textit{jump} distance. The \textit{Play as I rollover songs} tool instantly triggers the playback of the song closest to the cursor. A qualitative user study was undertaken with 10 subjects.

\footnotesize{Anita Shen Lillie. “MusicBox: Navigating the space of your music”. MA thesis. Massachusetts Institute of Technology, 2008}
In 52, Lübbers and Jarke created a 3D audiovisual environment to browse music songs heavily inspired by video games, up to its control device, an X60 console gamepad. A Self-Organizing Map (SOM) is used to represent the large feature space onto a 3D visualization, following very closely the metaphor of a geographical map; with sky, ground and sea textures rendered. Boxes representing songs with the album cover mapped onto each side float over the ground. Numerous audio features are extracted: RMS values of spectral centroids, roughness, key clarity, tempo estimates with the MIRtoolbox; stochastic MFCC models, fluctuation patterns and periodicity histograms; as well as similarity metrics from collaborative music recommendation portals: Last.FM53 and The Art of The Mix 54. Audio spatialization plays back all songs within the focus of perception from the view direction, with amplitudes attenuated as the distance grows from the focus, a sort of audio equivalent to the visual fisheye distortion. To adapt the visualization and underlying organization of songs, the user can build or destroy hills, this information being reused for optimizing the SOM. They evaluated two aspects of their design with 9 participants with collections of 100 tracks: the positioning of songs (versus a random placement) and the spatialized audio rendering (versus a standard media player).
2.2.6.13  SoundTorch by Heise et al. (2008)

SoundTorch has been designed by people aware of audio engineering practices. It relies on Mel-Frequency Cepstral Coefficients (MFCCs) as features, clustered with a Self-Organizing Map (SOM). To make maps comparable, the SOM is not initialized randomly as usual, but with an unclear geometric operation over groups of bins of the MFCCs. The authors claim that such an initialization ends up in mapping the horizontal axis correspond to a tonal-to-noisy continuum and the vertical axis to pitch increase or dull-to-bright. SoundTorch makes use of the node shape to convey additional information: the temporal evolution of the power of the signal is mapped to a circle. When the file is played back, its icon rotates so that the current playback location corresponds to the top of the circular representation. A Wii remote controller can be used through its accelerometer or infrared sensor to browse the space similarly to a flashlight. Spatialized sound rendering enriches the immersion. The authors positively evaluated their application through quantitative tests, know-item search tasks and described item tasks, with 15 users, compared to a list-based layout from Sony Vegas 8. It is not clear from this comparison whether SoundTorch outperforms the list-based application because of its content-based or interactive abilities, particularly its instant playback of closely-located nodes in the map.
SongExplorer by Julià and Jordà (2009) is a tangible user interface for browsing songs by content-based similarity. To organize 6666 songs of the Magnatune library, it relies on high-level features such as beats per minute and probability of the song to belong to some mood classes (happy, sad, party, aggressive, relaxed, acoustic) distributed on a 2D space by a Self-Organizing Map (SOM). A particularity of their application is that since the screen of the tabletop is circular, they adapted the SOM with a circular hexagonally-connected neuron grid, so that songs are mapped onto a circular view, and a restriction of only one song per neuron, so that distances can be better preserved and distributed. Each song is represented by a circle that pulses along the BPM and whose color corresponds to mood properties from mappings determined through online survey. Interacting with the application is finger-based (selection, zoom, translation) and tangible-based (pucks for color filtering, magnifying metadata, map navigation and playlist management). A qualitative experiment with an undisclosed number of subjects validated the design of the interface, particularly the color mappings.

2.2.6.15  **MuVis by Dias et al. (2010)**

Designed after an online survey and contextual inquiry, *MuVis* by Dias et al. \(^{58}\) proposes a combined semantic and content-based approach. It extracts audio features (*fluctuation patterns*) and summaries (*audio snippets*) with the CoMIRVA library. In addition to audio player (bubble 4) and playlist management (bubble 3) views, its user interface features two less common views using a spatially ordered tree map, one for filtering through queries (bubble 2), the other for their results or showing the overall collection (bubble 1). Possible queries, opted for based on the contextual inquiry results, are: plain text, musical similarity (song, album and artist), duration, year, genre, beat, and mood. 10 users performed 4 timed-based tasks (text-based search, browsing, exploration, similarity search) and were significantly faster than with the compared *Windows Media Player* except for one task. It must be stressed that this is the first research work chronologically from the content-based audio browsers category of this overview that reported results of proper statistics beyond means (Student tests with p-values). The source code is available under a GPLv3 license \(^{59}\).

---


\(^{59}\) [http://sourceforge.net/projects/fmuvis/](http://sourceforge.net/projects/fmuvis/)
Sonarflow by Lidy et al. \cite{Lidy2010} from Spectralminds, a company born out of researchers in multimedia information retrieval. Having previously worked on adapting Self-Organizing Maps (SOM) to early mobile devices (phones, pocket PCs...), it is unclear which techniques are used in SonarFlow. Sonarflow is an application for current mobile devices (multitouch tablets and phones), it organizes, in a 2D view following the metaphor of planets in outer space, genres, artists, albums and songs, each appearing depending on the level of zoom over the collection. Connecting to online recommender systems such as Last.fm and more recently Gracenote’s Spotify, it allows to discover new music with suggestions overlaid while navigating in personal collections.

\cite{Lidy2010} Thomas Lidy. “Sonarflow - Visual Music Exploration & Discovery”. In: Proceedings of ISMIR. 2010

For MusicGalaxy, Stober et al. chose the Landmark-based Multidimensional Scaling (LMDS) technique for reducing the dimensions of features: Kullback-Leibler divergence of Gaussian Mixture Models of Mel-Frequency Cepstral Coefficients, Euclidean distance of dynamics and fluctuation patterns, cosine distance of tf×idf-weighted terms of lyrics. A salient feature of their application is the SpringLens, a lens distortion technique that allows to separate overlapping items on the view by a non-linear distortion of the coordinates. This technique builds upon the fish-eye lens by adding other focii to the primary focus region of interest by computing neighborhoods, here content-based. The authors evaluated this technique quantitatively (task success) and qualitatively with 30 testers, opposed to and combined with pan and zoom, with a forked application named PhotoGalaxy loading databases of hundreds of images, which is somehow intriguing since the media type of the test application differs from the one of the described application, even if album covers are insightful for some music genres.

Users moderately made use of secondary focii since these were mainly invisible due to panning and zooming.


tf×idf: term frequency-inverse document frequency

2.2.7 Content-based video browsers

2.2.7.1 Informedia Digital Library Interface by Christel et al. (1995-2008)

Michael Christel and the Informedia group at Carnegie Mellon University have been pioneering the research on video browsing systems since the mid 1990’s. Their Informedia Digital Library Interface was tailored for mining in video news and documentaries and radio broadcasts. Though speech processing, image analysis and natural language processing, their system generated summaries such as headlines, film-strip storyboards and video skims, as illustrated in Figure 2.34, left screenshot. The second screenshot shows complementary visualizations among others: a scatter plot of hot topics, geographical maps for historical backgrounds and contexts.

Not only sustaining successful participations to TRECvid evaluation campaigns of content-based techniques, they pioneered into evaluating the usability of video browsers in a user-centered approach.

---


2.2.7.2 FutureViewer by Campanella et al. (2005)

*FutureViewer* by Campanella et al. ⁶⁸ is one among a few video browsers to sport a scatter plot, for instance the whole feature space of low-level MPEG-7 descriptors projected to 2D through a Principal Component Analysis (PCA) as illustrated in Figure 2.35 (left) or one feature assigned per axis (right), while most other video browsers opted for condensing at best rectangular video content in the available visual space, mostly in grids or storyboards. Below the plot, color bars represent the whole video by stripes corresponding to shots ordered temporally, of width determined by the shot duration and color by the shot dominant color.

Figure 2.35: Screenshots: PCA (left) and scatter plot (right)

<table>
<thead>
<tr>
<th>media</th>
<th>granularity</th>
<th>organization</th>
<th>visualization</th>
<th>interaction</th>
<th>users</th>
<th>experts</th>
<th>setting</th>
<th>desk</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.2.7.3 VideoSOM by Bärecke et al. (2006)

VideoSOM by Bärecke et al. 69 organizes video segments using a self-organizing map. Videos are first segmented into shots from rapid changes in color histograms in the Intensity;Hue;Saturation (IHS) space. Each shot is then represented by its median keyframe to proceed with feature extraction: in addition to the aforementioned global color histogram, histograms for cardinal regions of each keyframe are also extracted. To better maintain distances between adjacent items in the 2D map, they used hexagonal cells instead of the traditional rectangular topology for a self-organizing map (SOM) that reduces the dimensions of the feature space to the visual space. Each shot is represented by its keyframe in the visualization, positioned inside hexagonal cells by the SOM, the brightness of the background color (green) of each hexagon indicating the number of shots it holds in the overall view. Clicking on cells display the list of shot it contains. Selecting a shot plays back the video at its location. The background colors turn red and their brightness is then mapped to the distance between shots. The seek bar named timebar indicates most similar shots to the current also by color, overlaid, and the other shots from the same cell by rectangles.

2.2.7.4 ITEC Video Explorer by Schoeffmann et al. (2010)

In 70, Klaus Schoeffmann presented the *ITEC Video Explorer* tailored for browsing recently recorded video, on which time-consuming concept detection can not be applied on time, for instance: hot news, surveillance videos. Its user interface proposes a main video player view assorted with stackable seek bars that combine their overall and detail views, and that are extended with features extracted from the video, particularly HSV color and motion flow, and with frame-based visuals. Similarity search for similar scenes can be performed by selecting region of interests in displayed frames or in seeker bars. Shots can also be filtered by color (positively and negatively), cardinal motion directions, and camera zoom or ride.

In 71, the authors positively evaluated the motion layout summary against a standard seeker-bar video player, on 2 test videos, with 16 subjects, testing time to find a segment (quantitative) and appreciation through the System Usability Score (SUS) (qualitative). Schoeffmann founded since 2012 a live comparative user evaluation of video browsers named the Video Browser Showdown (VBS).

---


2.2.7.5 MediaMill by de Rooij et al. (2010)

MediaMill has been developed for a decade at the University of Amsterdam. It combines feature extraction and concept detection. Two visualization techniques named “thread”-based were presented in \(^72\): the CrossBrowser allows to browse two threads of content-based filtering queries, one vertical, the other horizontal, crossing each other at the original query. The RotorBrowser extends this concept by allowing more than two concurrent queries/threads, up to 8. Their evaluation with 32 students show that threads are beneficial for video search, but that users achieve better results with fewer threads: while the CrossBrowser is best suited for targeted search, the RotorBrowser favors exploratory search with its added search directions. More recently, they introduced a new iteration over their browsers: the ForkBrowser \(^73\). Allowing multiple threads like their predecessors, the ForkBrowser this time dedicates threads: the horizontal left and right threads form a timeline, the bottom vertical thread act as history of the queries, the top vertical thread announces the next results, the oblique threads propose visually similar candidates.
2.2.7.6 MediaTable by de Rooij et al. (2010)

MediaTable combines automatic content-analysis techniques based on MediaMill with a user interface optimized for categorizing visually massive sets of results. As illustrated in Figure 2.39, MediaTable offers different views. The Table stacks keyframes of shots in a spreadsheet with a condensed row lens effect, of height increased when hovering with the mouse. Columns correspond to the presence of detected concepts in the videos. Below the spreadsheet segments are or can be sorted in buckets: seen/unseen, selected/everything, user-defined. These buckets can be toggled to filter out the table, or be visualized on other views, such as the Grid, a matrix of thumbnails; or the Point Cloud, a scatter plot by assigning one column of the table to each axis.

They performed a quantitative evaluation with 12 teams of 2 members, measuring task success and times.
2.2.7.7 Galaxy Browser by Pang et al. (2011)

The Galaxy Browser acts as a search engine for videos. Items are represented in a galaxy, distributed in a force-directed graph representation, the similarity is computed from the textual information present as metadata or crawled from Wikipedia, Google News and Google Trends. Hyperlinks between videos are detected from partial near-duplicates and form constellations or visual snippets on certain topics, visualized through node links. A longer following-up paper mentions “usability” visibly as future work, after a performance evaluation solely algorithmic.

Figure 2.40: Screenshot

Table 2.42: Taxonomy


Bailer et al. contested to the 2012 Video Browser Showdown with a tool targeted to media post-production. Based on several features recommended by the MPEG-7 standard, such as camera motion and visual activity estimation, global color, object trajectories; hierarchical clustering allows users to refine the number of candidate segments to browse. Video segments are displayed on a light table, a grid of keyframes, with colors on bounding rectangles associated to clusters. A history log is available on the left hand to revert back to any previous step of the search process, and segments can be memorized on an edit decision list for further reuse (here media post-production).
Also following the MPEG-7 approach, the web-based **OVIDIUS** platform (for “On-line VIDeo Indexing Universal System”) by Bursuc et al.\(^7\) decomposes videos into scenes, shots and keyframes. Below a video player view, similar segments corresponding to the playback are displayed in rows, multiple rows being possible with different levels of granularity, similarly to the threads in **MediaMill** presented earlier. Each row from the visualization has its own pair of keys assigned for backward/forward navigation between segments. These key pairs belong to a distinctive row from the keyboard to make it convenient to memorize. One salient feature is that this tool allows to perform queries by selecting objects and regions of interests in frames. They also contested to the 2012 Video Browser Showdown for a quantitative evaluation.
With the 3D Thumbnail Ring, Schoeffmann et al. \(^{79}\) opted for sorting linearly sampled video keyframes by their dominant color on a cylinder of vertical revolution, the hue component from the Hue Saturation Value (HSV) color space mapped to the angle. The sequence of colors remains consistent with any video, even if the proportions change. The keyframes belonging to the same column (similar hue value) are sorted vertically so as to minimize the Euclidean distance between their HSV values. The rotation of the carrousel is mapped to the mouse wheel. They also participated in the 2012 Video Browser Showdown for a shared quantitative evaluation.

2.2.7.11 Graphic Object Searcher by Ventura et al. (2012)

The Graphic Object Searcher user interface by Ventura et al. is as well based on a visual representation in a grid. Videos are first processed offline to extract MPEG-7 color and texture features then indexed on a graph-based structure using the Hierarchical Cellular Tree (HCT) algorithm providing a multi-scale representation. This browser was a challenger in the 2012 Video Browser Showdown.

Figure 2.44: Screenshot

media
granularity
organization
visualization
interaction
users
setting
desk
community
availability
Java demo
usability (VBS 2012)

Table 2.46: Taxonomy

2.2.7.12  *HiStory by Hürst and Darzentas (2012)*

*HiStory* stands for *Hierarchical Storyboards*. Hürst and Darzentas \(^{81}\) designed this browser for mobile devices for known-item search tasks in single videos, extending a storyboard view (grid of thumbnails) that can be scrolled with a vertical slider informing of the temporal location and duration of the subset displayed on the grid in regard to the whole video. In contrast to other storyboard scrolling techniques they compared their design to, that are scrolling (1) continuously with the slider and (2) discretely page-by-page as in textual documents, they allowed the user to select a thumbnail so as to hierarchically recalculate the grid contents around the time frame of the chosen thumbnail. Their evaluation with 26 users show that their design didn’t outperform the two other scrolling techniques (1-2), suspected to be due to the hierarchical grid recompilation times, but still allowed testers to find targets in time for all tasks. They discuss as future work to replace the linear sampling of thumbnails with a content-based approach.

---

Very recently, Hudelist and Schoeffmann\textsuperscript{82} have introduced the \textit{3D filmstrip} browser tailored for mobile devices and single video files, maximizing the occupation of the screen by unrolling a film strip reminiscent of analog cinema. In its early stage, content-based techniques are not yet made use of in this browser, featuring face recognition is in planning. Keyframes are linearly sampled from the source video. Finger gestures allow to progress forward/backward over the swirls of the film strip and to adapt the granularity of the keyframe sampling.

2.2.7.14 Video Archive Explorer by Haesen et al. (2013)

The Video Archive Explorer by Haesen et al. 83, tailored for the needs of professionals of the TV broadcasting industry, was developed all along through a user-centered methodology: contextual inquiry, visual scenarios of use, iterative prototyping of increasing fidelity. It runs on a large multitouch display to allow collaboration. After a text-based query, results are displayed on screen in an animated slideshow, labelled with concepts detected automatically either regarding the story or associated to faces. Story segmentation is performed by mining from time-aligned subtitles and audiovisual cues. When selecting one result, it is displayed in the center of the video clock visualization, face- or story-related segments ordered clockwise around it. Another view, the video timeline visualization, aligns all the labelled keyframes on a timeline right under the selected result. The advanced video player is a video player sided with vertically-stacked annotation layers, similarly to video sequencers and annotation tools. The application has been evaluated with ten experts, qualitatively, concluding that most users would rather use a desktop version individually but the multitouch setup in meetings.

2.2.8 Freehand tools

2.2.8.1 Sonic Mapper by Scavone et al. (2002)

So as to validate content-based algorithms, an approach is to ask users to categorize media elements manually on a 2D space. The Sonic Mapper by Scavone et al. offers such an approach for sounds that can be positioned on user-defined regions from a map with possible sub-hierarchies. Additional pairwise similarity ratings submitted to users can help refine to organization. An important conclusion to note from the evaluation of such a tool is that testers would want to spend hours ordering collections of sounds notably because they could define their own ordering system or strategy and feel proud of their results.

Figure 2.48: Screenshot

Table 2.50: Taxonomy

---

2.2.9 Content-based artistic works

In ²⁵, we evaluated how scientific works on multimedia information retrieval, particularly content-based similarity, can foster new practices in digital or new media arts. A presentation of some of these works follows.

2.2.9.1 Pockets Full of Memories by George Legrady (2001-2007)

One artist and scientist that has been using such technologies as a theme for his artistic works is George Legrady ⁶⁶, notably since Pockets Full of Memories (2001-2007) ⁶⁷ an installation where participants can scan everyday objects which are then virtually organized in a projected Kohonen self-organizing map, based on pairwise ratings by the visitors of several aspects, thus proposing an emergent ordering since each individual induces her/his own perception of the object entered in the database that might differ from the visual similarities. This installation is featured in the book edited by Victoria Vesna ⁸⁸, among other similar works.

---


⁶⁶ http://www.georgelegrady.com


A subsequent work from George Legrady is *Cell Tango* (2006-2010), visualizing user-contributed cellphone-taken images by tags, in different visualizations. The *Cell_Bin* starts by positioning randomly the largest images and fill the spaces iteratively with the others. The *Cell_Clusters* selects the most recent images and fetches images similarly tagged from Flickr to surround these. The *Cell_Burst* is a variation animating the clusters with node links. The *Cell_Finale* displays the whole collection randomly positioned and sized, in a collage.
Martin Wattenberg’s *The Shape Of Song* (2001) proposes abstract visualizations in arc diagrams of the musical structure of hundreds of songs. Starting from MIDI transcripts of the musical pieces (sequences of notes defined by pitch, onset and duration), summaries are computed using the maximal matching pair algorithm and other rules to reduce the complexity of this algorithm. These summaries are visualized using overlaid semi-circular arcs whose thickness corresponds to the duration of the repeated musical passages, therefore incidentally underlining their relevance. Several non-interactive printouts of these arc diagrams have been exhibited as seen in Figure 2.51.

---

Figure 2.51: Picture taken at Generator.x, Oslo, 2006. Photo by Marius Watz (Creative Commons NC-BY-SA)

Martin Wattenberg’s *The Shape Of Song* (2001) proposes abstract visualizations in arc diagrams of the musical structure of hundreds of songs. Starting from MIDI transcripts of the musical pieces (sequences of notes defined by pitch, onset and duration), summaries are computed using the maximal matching pair algorithm and other rules to reduce the complexity of this algorithm. These summaries are visualized using overlaid semi-circular arcs whose thickness corresponds to the duration of the repeated musical passages, therefore incidentally underlining their relevance. Several non-interactive printouts of these arc diagrams have been exhibited as seen in Figure 2.51.

---

http://www.bewitched.com/song.html
2.2.9.4 Every Shot, Every Episode by Jennifer and Kevin McCoy (2001)

Artist couple Jennifer and Kevin McCoy manually populated a database of every shot from twenty episodes of the television show *Starsky and Hutch*. They recorded this material on 300 CDs that were installed on shelves at exhibitions, as seen in Figure 2.52, each representative of formal, visual and narrative elements, for instance: “every car chase”. Open briefcases positioned over the shelf and holding a small display connected to a disc player invited visitors to browse the collection. Without much help from current multimedia technologies, these manual segmentation and concept detection tasks must have been quite time-consuming and burdensome.

http://www.mccoyspace.com
2.2.9.5 *cinemetrics by Frederic Brodbeck (2011)*

For his Bachelor in Arts at the Royal Academy of Arts (KABK), Den Haag, Frederic Brodbeck created *cinemetrics*[^91], animated fingerprints of movies obtained from analysis of DVD data (audio and video channels, subtitles, chapters) and quotes from the Internet Movie Database (IMDB). A nice outcome from his work are posters featuring several of his visualizations that he offered for sale as collectibles. As he released his source code[^92], this design is of inspiration and could be reused for the glyphs representing elements of a collection of movies in browsers.

[^91]: http://cinemetrics.fredericbrodbeck.de

[^92]: https://github.com/freder/cinemetrics/
2.3 Organizing these browsers

In this section we isolate or combine traits of the taxonomy described beforehand to provide different layer of analysis of the state-of-the-art of media browsers, again mostly focused on content-based ones. Table 2.56 compares browsers by media type (🎵 musica 🎥 video 📽️ film 🌐 web 📲 phone), granularity ( örg organisé), organization (ër organisé), users (قواطع), communities (عناصر) and usability evaluation (الميزات).

2.3.1 Users, media types, granularity

These audio and video browsers target different user categories: most audio browsers presented here are for personal music collections, whereas video browsers are targeted to professionals (video surveillance, news and broadcast specialists). Tools for sound design and personal movie collections have been less studied as research works.

2.3.2 Communities

Research on content- and context-based browsers seems to be bound respectively to the MIR and HCI communities: the former have been presented mostly in multimedia information retrieval conferences; whereas the latter target human-computer interaction conferences. However, both communities have a common root: computers. Recently, there has been a dispute in France whether human-computer interaction should be considered a part of signal processing or computer science. Interdisciplinary collaborations would help make both practices benefits meet, as done by the art community. A trend seems to emerge with recent browsers, since 2010, fields are converging, faster for video content.

2.3.3 Usability evaluations

In her PhD thesis including her work around the amblr browser analyzed earlier in this chapter, Stewart shows that published works on applications with auditory display, including audio browsers that have been presented here, tend to provide inaccurate usability evaluations, ever since the HCI and MIR communities specialized and split in the early current century. Here we follow up on her work studying audiovisual media browsers. The choice of sample sizes or minimal numbers of users for trustable evaluations is still unclear today: is a rule of thumb to spot 80% of common usability issues, 16 for qualitative evaluations to be statistically significant.

93 Rebecca Stewart. “Spatial Auditory Display for Acoustics and Music Collections”. PhD thesis. School of Electronic Engineering and Computer Science, Queen Mary, University of London, 2010

<table>
<thead>
<tr>
<th>Browser</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Forager</td>
<td>1996</td>
</tr>
<tr>
<td>Raskin</td>
<td>2001</td>
</tr>
<tr>
<td>BumpTop</td>
<td>2006</td>
</tr>
<tr>
<td>Liquifile</td>
<td>2004</td>
</tr>
<tr>
<td>Film Finder</td>
<td>1993</td>
</tr>
<tr>
<td>AAU Video Browser</td>
<td>2010</td>
</tr>
<tr>
<td>Panopticon</td>
<td>2013</td>
</tr>
<tr>
<td>Sonic Browser</td>
<td>2001</td>
</tr>
<tr>
<td>Strands</td>
<td>2004</td>
</tr>
<tr>
<td>The amblr</td>
<td>2008</td>
</tr>
<tr>
<td>AudioFinder</td>
<td>1986</td>
</tr>
<tr>
<td>Soundminer HD</td>
<td>2002</td>
</tr>
<tr>
<td>Adobe Bridge</td>
<td>2005</td>
</tr>
<tr>
<td>Vintage Radio Interface</td>
<td>2012</td>
</tr>
<tr>
<td>Greenstone</td>
<td>2000</td>
</tr>
<tr>
<td>Bohemian Bookshelf</td>
<td>2011</td>
</tr>
<tr>
<td>SpeechSkimmer</td>
<td>1993</td>
</tr>
<tr>
<td>SoundFisher</td>
<td>1996</td>
</tr>
<tr>
<td>Marsyas3D</td>
<td>2001</td>
</tr>
<tr>
<td>SmartMusicKIOSK</td>
<td>2003</td>
</tr>
<tr>
<td>Musicream</td>
<td>2005</td>
</tr>
<tr>
<td>MusicMiner</td>
<td>2005</td>
</tr>
<tr>
<td>Search Inside The Music</td>
<td>2007</td>
</tr>
<tr>
<td>Music Rainbow</td>
<td>2006</td>
</tr>
<tr>
<td>Globe of Music</td>
<td>2007</td>
</tr>
<tr>
<td>MusicBox</td>
<td>2008</td>
</tr>
<tr>
<td>soniXplorer</td>
<td>2009</td>
</tr>
<tr>
<td>SoundTorch</td>
<td>2008</td>
</tr>
<tr>
<td>SongExplorer</td>
<td>2009</td>
</tr>
<tr>
<td>MuVis</td>
<td>2010</td>
</tr>
<tr>
<td>Sonarflow</td>
<td>2010</td>
</tr>
<tr>
<td>MusicGalaxy</td>
<td>2010</td>
</tr>
<tr>
<td>CataRT</td>
<td>2004</td>
</tr>
<tr>
<td>Informedia DLI</td>
<td>1995</td>
</tr>
<tr>
<td>FutureViewer</td>
<td>2006</td>
</tr>
<tr>
<td>VideoSOM</td>
<td>2006</td>
</tr>
<tr>
<td>ITEC Video Explorer</td>
<td>2010</td>
</tr>
<tr>
<td>MediaMill</td>
<td>2010</td>
</tr>
<tr>
<td>MediaTable</td>
<td>2010</td>
</tr>
<tr>
<td>Galaxy Browser</td>
<td>2011</td>
</tr>
<tr>
<td>Johaneum Browser</td>
<td>2012</td>
</tr>
<tr>
<td>OVIDIUS platform</td>
<td>2012</td>
</tr>
<tr>
<td>3D Thumbnail Ring</td>
<td>2012</td>
</tr>
<tr>
<td>Graphic Object Searcher</td>
<td>2012</td>
</tr>
<tr>
<td>HiStory</td>
<td>2012</td>
</tr>
<tr>
<td>3D Filmstrip</td>
<td>2013</td>
</tr>
<tr>
<td>Video Archive Explorer</td>
<td>2013</td>
</tr>
<tr>
<td>Sonic Mapper</td>
<td>2002</td>
</tr>
<tr>
<td>Pockets Full Of Memories</td>
<td>2001</td>
</tr>
<tr>
<td>Cell Tango</td>
<td>2006</td>
</tr>
<tr>
<td>Shape of Song</td>
<td>2001</td>
</tr>
<tr>
<td>Every Shot Every Episode</td>
<td>2001</td>
</tr>
<tr>
<td>cinemetrics</td>
<td>2011</td>
</tr>
</tbody>
</table>

Table 2.56: Browsers comparison
Table 2.57 compares browsers by media type (Audio, Video), granularity (Granularity), organization (Organization), visualization dimensions (Visualization) and interaction (Interaction).

2.3.4 Visualization, media types, granularity

The first striking evidence that emerge from the comparison of these audio and video browsers is that their designers have been creative in trying different algorithms 1) to position audio segments or files in a 2D space and 2) to convey their content through visual node representations. However, the challenges are different for audio and video. First, in contrast to video content, audio content is not visual by nature. So there is a need to complement this gap by providing condensed visual representations of audio signals. Second, from the contraposition: since video content is visual, the most space-effective representation and organization of several items in a 2D space is a grid, of granularity dependent of the video dimensions.

A complementary remark is that collection visualization may be adapted whether the user aims at discovering the structure of unknown databases and stumbling upon unexpected material (“serendipity”); or for performing intended search with determined purposes.

2.3.5 Interaction, settings

Most research on content-based browsers seems to have been targeted to WIMP interfaces, still default for personal and expert computers. Mobile solutions are emerging for everyday personal use, and tabletops for collaborative expert use. Would alternative user interfaces be of interest for expert users in offices, more engaging for long hours of work than keyboards and mice? Or for collective experience in public settings?
<table>
<thead>
<tr>
<th>Browser</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Forager</td>
<td>1996</td>
</tr>
<tr>
<td>Raskin</td>
<td>2001</td>
</tr>
<tr>
<td>BumpTop</td>
<td>2006</td>
</tr>
<tr>
<td>Liquifile</td>
<td>2004</td>
</tr>
<tr>
<td>Film Finder</td>
<td>1993</td>
</tr>
<tr>
<td>AAU Video Browser</td>
<td>2010</td>
</tr>
<tr>
<td>Panopticon</td>
<td>2013</td>
</tr>
<tr>
<td>Sonic Browser</td>
<td>2001</td>
</tr>
<tr>
<td>Strands</td>
<td>2004</td>
</tr>
<tr>
<td>The amblr</td>
<td>2008</td>
</tr>
<tr>
<td>AudioFinder</td>
<td>1986</td>
</tr>
<tr>
<td>Soundminer HD</td>
<td>2002</td>
</tr>
<tr>
<td>Adobe Bridge</td>
<td>2005</td>
</tr>
<tr>
<td>Vintage Radio Interface</td>
<td>2012</td>
</tr>
<tr>
<td>Greenstone</td>
<td>2000</td>
</tr>
<tr>
<td>Bohemian Bookshelf</td>
<td>2011</td>
</tr>
<tr>
<td>SpeechSkimmer</td>
<td>1993</td>
</tr>
<tr>
<td>SoundFisher</td>
<td>1996</td>
</tr>
<tr>
<td>Marsyas3D</td>
<td>2001</td>
</tr>
<tr>
<td>SmartMusicKIOSK</td>
<td>2003</td>
</tr>
<tr>
<td>Musicream</td>
<td>2005</td>
</tr>
<tr>
<td>MusicMiner</td>
<td>2005</td>
</tr>
<tr>
<td>Search Inside The Music</td>
<td>2007</td>
</tr>
<tr>
<td>Music Rainbow</td>
<td>2006</td>
</tr>
<tr>
<td>Globe of Music</td>
<td>2007</td>
</tr>
<tr>
<td>MusicBox</td>
<td>2008</td>
</tr>
<tr>
<td>soniXplorer</td>
<td>2009</td>
</tr>
<tr>
<td>SoundTorch</td>
<td>2008</td>
</tr>
<tr>
<td>SongExplorer</td>
<td>2009</td>
</tr>
<tr>
<td>MuVis</td>
<td>2010</td>
</tr>
<tr>
<td>Sonarflow</td>
<td>2010</td>
</tr>
<tr>
<td>MusicGalaxy</td>
<td>2010</td>
</tr>
<tr>
<td>CataRT</td>
<td>2004</td>
</tr>
<tr>
<td>Informedia DLI</td>
<td>1995</td>
</tr>
<tr>
<td>FutureViewer</td>
<td>2006</td>
</tr>
<tr>
<td>VideoSOM</td>
<td>2006</td>
</tr>
<tr>
<td>ITEC Video Explorer</td>
<td>2010</td>
</tr>
<tr>
<td>MediaMill</td>
<td>2010</td>
</tr>
<tr>
<td>MediaTable</td>
<td>2010</td>
</tr>
<tr>
<td>Galaxy Browser</td>
<td>2011</td>
</tr>
<tr>
<td>Johaneum Browser</td>
<td>2012</td>
</tr>
<tr>
<td>OVIDIUS platform</td>
<td>2012</td>
</tr>
<tr>
<td>3D Thumbnail Ring</td>
<td>2012</td>
</tr>
<tr>
<td>Graphic Object Searcher</td>
<td>2012</td>
</tr>
<tr>
<td>HiStory</td>
<td>2012</td>
</tr>
<tr>
<td>3D Filmstrip</td>
<td>2013</td>
</tr>
<tr>
<td>Video Archive Explorer</td>
<td>2013</td>
</tr>
<tr>
<td>Sonic Mapper</td>
<td>2002</td>
</tr>
<tr>
<td>Pockets Full Of Memories</td>
<td>2001</td>
</tr>
<tr>
<td>Cell Tango</td>
<td>2006</td>
</tr>
<tr>
<td>Shape of Song</td>
<td>2001</td>
</tr>
<tr>
<td>Every Shot Every Episode</td>
<td>2001</td>
</tr>
<tr>
<td>cinematics</td>
<td>2011</td>
</tr>
</tbody>
</table>

Table 2.57: Browsers comparison
2.4 Problems to solve, requirements

Klaus Schoeffmann, co-author of several video browsers presented beforehand, provided a state on the art on video browsers and summarization techniques in his PhD thesis. He listed the following concerns to look upon as requirements to allow immediate video exploration: fast content analysis, an interactive and usable user interface, efficient metadata storage, define videos beyond shots. In the founders, including himself, and participants of the Video Browser Showdown, provide a review on video browsing applications.

Are there patterns for designing multimedia information retrieval interfaces? In define three stages for the video retrieval process ("query", "inspect", "navigate") and nine guidelines to design video browsers accordingly ("inspect", "temporal context", "multiple shots", "show unexplored", "default navigation", "inspect navigation options", "return to previous states", "no panes", "mapping between navigation and visualization").

2.4.1 Usability

2.4.1.1 Understanding how users (want to) sort their collections

Do users need content-based browsers? Rodden and Wood have interviewed a dozen of users and followed their use of a digital photo manager during 6 months in 2000 to understand how people sort their personal image collections. They noticed that users didn’t make use very often of advanced multimedia processing features such as content-based image retrieval. It is to be noted that the photo manager application used for the test, AT&T’s Shoebox, didn’t feature face recognition at the time.

Do users layout files in structures? Recently, Bergman et al. have surveyed how 296 people would sort their personal files: these favored breadth (the number of folders in a given folder) over depth (the number of parent folders of a given file or folder), with on average 2.86 folders of depth and 10.64 of breadth, with 11.82 files per folder. Such a structure proved to be efficient: 94% targets could be found in 14.76s in 1131 tasks.

96 Klaus Schoeffmann, Frank Hopfgartner, Oge Marques, Laszlo Boeszoermenyi, and Joemon M. Jose. “Video browsing interfaces and applications: a review”. In: SPIE Reviews 1, 018004 (2010), pp. 1–35. doi: 10.1117/6.0000005
2.4.1.2 Focusing on sound designers

Sound designers organize their sound libraries either with dedicated applications (often featuring spreadsheet views), or with default file browsers. Sound designers source sounds in massive collections, heavily tagged by themselves and sound librarians. If a set of sounds to compose the desired sound effect is not available, a Foley artist (that can be the sound designer herself/himself) records the missing sound and tags these recordings as accurately as possible, identifying many physical (object, source, action, material, location) and digital (effects, processing) properties. When it comes to looking for sounds in such collections, successive keywords can help the user to filter down the results. But at the end of this process, hundreds of sounds can still remain for further review. This creates an opportunity for content-based information retrieval approaches and other means for presenting the available content.

2.4.1.3 Facilitating content-based browsing for non MIR specialists

Most of the tools described so far are often designed by multimedia information retrieval specialists that have an expert knowledge of the underlying processes and workflows. But how about novice users? Should they be offered intricate browsing capabilities such as choosing the combination of features for the clustering, based on the “cryptic” meaning that is induced by their names? Perceptual features such as loudness can be self-explanatory. How about “spectral skewness”?

2.4.2 Information visualization

2.4.2.1 Sustaining mental models

A rule of thumb from the human perception and cognition is that people won’t track or follow more than five elements or events in a visual scene. Board games such as Memory challenges the players to sustain their mental model of visuals of a grid of cards flipped upside-down by revealing 2 of them at a time. Similarly, in a 2D space representing a collection of media content, how can the user progressively remember and learn the space, especially for non-visual media types such as audio and when browsing parameters can be tweaked?
2.4.2.2 Favoring 2D visualizations versus 3D

Content-based analysis can produce a large set of high-dimensional features. Dimension scaling to the visible spaces drastically reduces the information to 3 or 2 dimensions. What are the gains of keeping an added dimension in 3D views? With planar display devices, occlusions may occur in 3D spaces, so the user must compensate by manually panning or rotating the view.

In 100, Hoashi et al. compared 2D and 3D visualizations for music information retrieval, particularly 2D clouds versus lists, and 2D versus 3D clouds, their evaluation seems to show that more dimensions improve the effectiveness of the user interface. However, in 101, Hearst illustrates by numerous research works that text lists in most cases outperform 2D scatter plots, themselves providing faster search than 3D interfaces. Is it related to the choices taken by the designers when mapping meaning to dimensions?

2.4.2.3 Mapping browsing cues to visual variables

How to best associate browsing cues to visual variables? Should position and distance always convey similarity? In 102, Hearst reminds from the guidelines provided by the Gestalt principles that a meaning of proximity is well conveyed through the closeness of objects in space. From the browsers presented in this chapter, only a few go beyond positions of media items, for instance SoundTorch adapted node color and contour to transmit more information on audio content. What is the individual influence of each of these features mapped to visual variables?

2.4.3 Interactivity

The majority of browsers we presented were designed for WIMP user interfaces (for “Window, Icon, Menu, Pointer”), a few added off-the-shelf devices, some opted for tabletops and multitouch user interfaces. Touch-based user interfaces have recently regain more interest due to the widespread use of multitouch mobile devices. But does it scale back to desk-based settings?
2.5 Conclusion

We provided a state-of-the art of media browsers, mostly for audio and video media types, relying on context- or content-based organization, using various solutions from multimedia information retrieval, information visualization and user interface design, created by several communities. After two decades of active research in these fields, very few of the content-based browsers passed the stage of being state-of-the-art for expert or novice media practitioners.

Regarding audio browsers, much work has been centered on music song collections, very few on sample collections tailored for sound designers. On contrary, most video browsers have been targeted to specialists (broadcast, surveillance), browsing personal movie collections is still not very user-friendly.

There is still room left for improvements: bridging the gap between these communities, deepening and generalizing the usability evaluation of these tools (notably for their comparison).
3 Method: interaction, organization

ONLY THOUGHT CAN RESEMBLE. IT RESEMBLES BY BEING WHAT IT SEES, HEARS, OR KNOWS; IT BECOMES WHAT THE WORLD OFFERS IT.

RENÉ MAGRITTE ¹

Navigating in directories and folders of files is an everyday task for people working with computers. File browsers are one of the standard applications expected to be available as default in desktop environments. These generally order files in lists or grids or trees by folder then filename, and offer spreadsheet views to refine the sorting by subsequent characteristics, extracted from the metadata of files (for instance date, author, file size) or mined from the document (when performing textual queries over text-based documents). For some decades, we’ve been accustomed to use such applications in a desktop situation, with displays sized like large books or newspapers, using keyboards and mice.

How about audio and video files? Is metadata sufficient to classify these? Are grids or lists the most efficient techniques to present collections of such files? What are the interaction techniques favored by experts manipulating audio and video content that preceded keyboards and mice before computerized systems? How about newer interaction techniques? How does it scale to recent device sizes, between mobility and large public spaces? These are the questions we address in this Chapter.

In section 3.1, we report on the state-of-the-art of a method which is complementary to metadata for the organization of media content: that is content-based similarity. In section 3.2, we provide an overview on visualization techniques for: information, time, and media content. In section 3.3, we focus on gestural input and control. In section 3.4, we give a short focus on rapid prototyping tools that can aid us developing our systems.

¹ Michel Foucault. “This is not a pipe”. In: Letter from May 23, 1966. University of California Press, 1983. Chap. Two letters by René Magritte
3.1 Organizing media content by content-based similarity

Metadata aims at providing a fair representation of data content with a few digits or several words. Is it sufficient to accurately describe complex content? When metadata is not enough, content-based similarity attempts to extract more characteristics from the recorded signal which constitutes a single element of media content. These characteristics, generally called features, can be considered as a simplified and comparable signature in regard to the original content. We will provide an overview of these for audio and video content in section 3.1.1. However sparse these can be, their dimension, or number of digits that form their representation, can still be too high to allow to sort them in dimensions humans can comprehend through their perception and senses, down to 2D and 3D: common dimension reduction methods will be summarized in section 3.1.4. Figure 3.1 summarizes how all these content-based processing steps can be sequenced in order to produce intra- and inter-media representations. By representation we mean knowledge that can be visualized or understood as a structure.
3.1.1 Feature extraction

Media content can be described through the many dimensions of human perception, traditionally though temporal and spatial features depending on the media type.

3.1.1.1 Audio

For sounds, low-level features can be temporal and spectral, some of these perceptual (influenced by the human perception mechanisms). For music, higher-level subcategories such as rhythm, melody and harmony are often borrowed. Geoffroy Petters’ report \(^2\) is one comprehensive and concise guide to audio features, which he orders therein into the following categories: temporal, spectral, based on energy, harmonic, and perceptual. While we refer to the aforementioned report for further information, let’s name a few among the most used features:

- Mel-Frequency Cepstrum Coefficients (MFCC) have been designed for speech analysis, but have been ever since of widespread use beyond, for music information retrieval. MFCCs are a measure of the spectral envelope. Most content-based audio browsers surveyed in our state-of-the-art on browsers Chapter \(^2\) made use of MFCCs.

- Spectral Flatness quantifies whether there are discontinuities in the spectrum.

- Loudness describes how loud sounds perceptually feel.

Thomas Grill \(^3\) aimed at defining high-level features correlated to perceived characteristics of sounds that can be named or verbalized through personal constructs, for instance high-low, ordered-chaotic, natural- artificial, smooth-coarse, tonal-noisy, homogeneous-heterogeneous. One application of this work is to simplify the user interface of MIR systems by making the choice of features more understandable to users not expert in signal processing.

\(^{2}\) G. Peeters. A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project. Tech. rep. IRCAM, 2004

3.1.1.2 Video

For video, researchers usually learn about the state-of-the-art starting with the features available in the OpenCV computer vision library, and end up improving these or developing new ones in the same framework. Common visual features are: color, shape, texture, optical flow. There are implementations of the ISO MPEG-7 standard for video features (ISO/IEC 15938:2001-15938:2004), for instance the official implementation by Bailer et al.

Song et al. investigated which audiovisual features are the most salient for the production of video summaries. For instructional documentary videos, summaries are guided by the audio channels and supported by the visual channel. Human faces and natural scenes are the most recurrent candidates from the visual channel as a source to craft video summaries. Regarding audio channels, human voices of isolated locators and natural sounds are often the base for these summaries.

3.1.2 Segmentation

While computing features aims at providing a signature to understand and compare media files, multiple dimensions are created, temporal and/or spatial depending on the media type, and mathematical dimensions depending on the features. To reduce the latter, statistics are usually applied: a mean is the most basic approach but it is quite restrictive when the content of a given file is not homogeneous, variance and standard deviation may applied, as well as higher order statistics such as kurtosis and skewness. A way to counterbalance this lack of homogeneity is to segment files along their temporal or spatial dimensions into chunks that are more representative when taken separately than over the whole file.

Roads has illustrated how different time scales can structure the understanding of music: from large scales such as the life of the composer and different interpretations of a song; to smaller scales such as verses, notes, textures, grains, buffers and samples. This conceptualization can be extended to visual media.

---


3.1.3 Distances and metrics

To quantify the notion of similarity between media elements of a collection, metrics based on distances can be defined and computed pairwise along chosen dimensions or features that are extracted from each element. Figure 3.2 adapted from \(^8\) visualizes three common metrics: the contours of the shapes define the equidistant path to the center node. The Euclidean distance is generally favored.

$$\sum_{i=0}^{d-1} |Q_i - P_i|$$

Manhattan (L\(_1\))

$$\sqrt{\sum_{i=0}^{d-1} (Q_i - P_i)^2}$$

Euclidean (L\(_2\))

$$\max(|Q_i - P_i|)$$

Maximum (L\(_\infty\))

where \(P_i (Q_i)\) is the coordinate of point \(P\) (Q) in dimension \(i\).

Weighted distances compensate the effects of scales between dimensions. For instance, the Mahalanobis distance corresponds to a weighted Euclidean distance:

$$\sqrt{\sum_{i=0}^{d-1} (\frac{Q_i - P_i}{s_i})^2}$$

where \(s_i\) the standard deviation between \(Q_i\) and \(P_i\).

3.1.4 Dimension reduction

Dimension reduction aims at providing a human readable understanding of a media collection by trying to decrease the many dimensions extracted features can offer. Here follow some criteria that dimension reduction should address and help choose of a method \(^9\).

- **Continuity**, a measure that can intuitively be related to recall in information retrieval, qualifies the number of high-dimensional neighbors that are preserved in low dimension.
- **Trustworthiness** (related to precision), represents the number of low-dimensional neighbors that relate in high dimension.
- **Overlap** occurs when the distance between several elements in low dimension is too small.
- **Perplexity** is the number of effective nearest neighbors.

\(^8\) Christian Böhm, Stefan Berchtold, and Daniel A. Keim. “Searching in High-dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases”. In: ACM Comput. Surv. 33.3 (Sept. 2001), pp. 322–373. ISSN: 0360-0300. DOI: 10.1145/502807.502809

The following methods are often cited or employed.

- Multidimensional Scaling (MDS) and Principal Components Analysis (PCA) determine the linear mixture of dimensions that best describe the contents of a dataset. MDS uses a distance matrix whereas PCA uses a correlation matrix.

- Self-Organizing Maps (SOM) and Growing Self-Organizing Map (GSOM) are inspired by neural processes, similar to Artificial Neural Networks, and are trained to optimize the weights of their neurons.

- Student-t distributed Statistical Neighbor Embedding (tSNE) and Neighbor Retrieval Visualizer (NeRV) are probabilistic methods that aim at preserving high-dimensional neighbors in lower-dimensional projections.

Some of these methods were compared by Nybo et al.\textsuperscript{11}:

- SOMs produced the most trustworthy projection.
- NeRV was superior in terms of continuity and recall.

Qualitative evaluations of different approaches on sound collections have been undertaken by Stober et al.\textsuperscript{12}. Users described that:

- MDS resulted in less positional changes;
- NeRV better preserved cluster structures;
- GSOM produced less overlapping.

Table 3.1 summarizes the dimension reduction techniques employed among the content-based inter-media browsers from our state-of-the-art. Most authors chose either PCA or SOMs.


<table>
<thead>
<tr>
<th>Browser</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Forager</td>
<td>1996</td>
</tr>
<tr>
<td>Raskin</td>
<td>2001</td>
</tr>
<tr>
<td>BumpTop</td>
<td>2006</td>
</tr>
<tr>
<td>Liquiffle</td>
<td>2004</td>
</tr>
<tr>
<td>Film Finder</td>
<td>1993</td>
</tr>
<tr>
<td>AAU Video Browser</td>
<td>2010</td>
</tr>
<tr>
<td>Panopticon</td>
<td>2013</td>
</tr>
<tr>
<td>Sonic Browser</td>
<td>2001</td>
</tr>
<tr>
<td>Strands</td>
<td>2004</td>
</tr>
<tr>
<td>The amblr</td>
<td>2008</td>
</tr>
<tr>
<td>AudioFinder</td>
<td>1986</td>
</tr>
<tr>
<td>Soundminer HD</td>
<td>2002</td>
</tr>
<tr>
<td>Adobe Bridge</td>
<td>2005</td>
</tr>
<tr>
<td>Vintage Radio Interface</td>
<td>2012</td>
</tr>
<tr>
<td>Greenstone</td>
<td>2000</td>
</tr>
<tr>
<td>Bohemian Bookshelf</td>
<td>2011</td>
</tr>
<tr>
<td>SpeechSkimmer</td>
<td>1993</td>
</tr>
<tr>
<td>SoundFisher</td>
<td>1996</td>
</tr>
<tr>
<td>Marsyas3D</td>
<td>2001</td>
</tr>
<tr>
<td>SmartMusicKIOSK</td>
<td>2003</td>
</tr>
<tr>
<td>Musicream</td>
<td>2005</td>
</tr>
<tr>
<td>MusicMiner</td>
<td>2005</td>
</tr>
<tr>
<td>Search Inside The Music</td>
<td>2007</td>
</tr>
<tr>
<td>Music Rainbow</td>
<td>2006</td>
</tr>
<tr>
<td>Globe of Music</td>
<td>2007</td>
</tr>
<tr>
<td>MusicBox</td>
<td>2008</td>
</tr>
<tr>
<td>soniXplorer</td>
<td>2009</td>
</tr>
<tr>
<td>SoundTorch</td>
<td>2008</td>
</tr>
<tr>
<td>SongExplorer</td>
<td>2009</td>
</tr>
<tr>
<td>MuVis</td>
<td>2010</td>
</tr>
<tr>
<td>Sonarflow</td>
<td>2010</td>
</tr>
<tr>
<td>MusicGalaxy</td>
<td>2010</td>
</tr>
<tr>
<td>CataRT</td>
<td>2004</td>
</tr>
<tr>
<td>Informedia DLI</td>
<td>1995</td>
</tr>
<tr>
<td>FutureViewer</td>
<td>2006</td>
</tr>
<tr>
<td>VideoSOM</td>
<td>2006</td>
</tr>
<tr>
<td>ITEC Video Explorer</td>
<td>2010</td>
</tr>
<tr>
<td>MediaMill</td>
<td>2010</td>
</tr>
<tr>
<td>MediaTable</td>
<td>2010</td>
</tr>
<tr>
<td>Galaxy Browser</td>
<td>2011</td>
</tr>
<tr>
<td>Johaneum Browser</td>
<td>2012</td>
</tr>
<tr>
<td>OVIDIUS platform</td>
<td>2012</td>
</tr>
<tr>
<td>3D Thumbnail Ring</td>
<td>2012</td>
</tr>
<tr>
<td>Graphic Object Searcher</td>
<td>2012</td>
</tr>
<tr>
<td>HiStory</td>
<td>2012</td>
</tr>
<tr>
<td>3D Filmstrip</td>
<td>2013</td>
</tr>
<tr>
<td>Video Archive Explorer</td>
<td>2013</td>
</tr>
<tr>
<td>Sonic Mapper</td>
<td>2002</td>
</tr>
<tr>
<td>Pockets Full Of Memories</td>
<td>2001</td>
</tr>
<tr>
<td>Cell Tango</td>
<td>2006</td>
</tr>
<tr>
<td>Shape of Song</td>
<td>2001</td>
</tr>
<tr>
<td>Every Shot Every Episode</td>
<td>2001</td>
</tr>
<tr>
<td>cinematics</td>
<td>2011</td>
</tr>
</tbody>
</table>

Table 3.1: Browsers compared: (content-based) organization, visualization
3.1.5 Frameworks for multimedia information retrieval

In Chapter 2 we reviewed audio and video browsers based on content similarity. Some of these applications were built upon frameworks for multimedia (information retrieval), for instance audio browsers like Marsyas 3D and Sonic Browser using Marsyas by Tzanetakis\(^{13}\), MuVis using CoMIRVA by Schedl et al.\(^{14}\) and most video browsers relying on OpenCV. The first two frameworks offer bindings in various languages, what can satisfy both developers and scientists: C++, Java, Python, and Matlab.

The Essentia framework by Bogdanov et al.\(^{15}\) is a powerful library, archiving and renewing many MIR algorithms that appeared in the last year of this thesis.

Other browsers rely on audio recommender systems such as Echo Nest, Last.FM and MusicBrainz that provide an API to developers to enable queries on metadata and/or similar artists/tracks from third party applications. When users can’t develop software prototypes by themselves, they can rely on audio editors such as Sonic Visualiser and Audacity that support plugins, notably through the VAMP SDK, with which parts of the content-based dataflow can be reproduce: computing spectrograms, extracting features.

Is there a need to create yet another framework?

Some of these libraries or frameworks are released under open source licenses that may be restrictive for commercial use, for instance GPL enforcing the right of the user to access the source code of the whole work derived and/or depending on the given library or framework. Yet Another Audio Feature Extraction by Mathieu et al.\(^{16}\) is a rare exception that is released under the less restrictive LGPL license (allowing to link closed-source applications against it).

Some of these libraries or frameworks run in the Matlab environment for engineering which is really quite not suitable for designing interactive applications.


\(^{16}\) Benoit Mathieu, Slim Essid, Thomas Fillon, Jacques Prado, and Gaël Richard. “YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software”. In: Proceedings of the 11th ISMIR conference. 2010
3.2 **Visualization: information, time, media content**

Feedback from computerized systems is primarily channeled through vision (for non-impaired users). In this section we will explore how information, time and media content can be visualized, displayed.

Card et al.’s book provides a repertory of information visualization techniques that laid the grounds of this discipline, with paradigms such as overview+detail and focus+context \(^{17}\).

3.2.1 **Design rules induced from human perception**

Bertin \(^{18}\) defined seven visual variables as a basis for visual communication, initially in the field of cartography. Such variables were ranked in groups revealing their capacity of conveying nominal, ordinal, and quantitative data, these orderings have been validated through user experiments by Cleveland et al. \(^{19}\) applied to data visualization. These visual variables have ever since been employed as a framework into the field of information visualization to assist the design of graphics in order to best transmit the proper information to the eyes depending on the use case \(^{20}\). Ware’s book goes further into detail \(^{21}\).

The analysis of the design of the audio waveform viewer based on preliminary work \(^{22}\) and pictured in figure 3.3 can be based upon these visual variables and serve as illustration to explain their usefulness.

![Audio waveform viewer](numediart)
• **Position** - The playback view is centered around a fixed vertical bar through which the data is flowing; whereas a moving bounding box surrounding the playback neighborhood within a wave summary that is fixed underlines the control flow;

• **Form** - The vertical bar of the playback view, that resembles a non-moving plot axis, tends to lock the center of the user’s viewpoint to the current sample; while the bounding box of the navigation view denotes the delimitation of the playback area of interest;

• **Orientation** - We followed a time axis orientation that is commonly horizontal and pointing towards the right in scientific graphs, while we ordered vertically the navigation view below the playback view more arbitrarily, as this issue appears to be more motivated by cultural issues (such as the top-to-down, left-to-right reading order in western societies);

• **Color** - Using the same color for the playback view and for the corresponding snapshot on the wave summary helps the user identify the two. Practically, we have re-ordered the color palette of the numediart logo by saliency even if we could have opted for a more artistic color composition \(^23\): the red color of the bounding box retains attention, the white waveform section of the playback region is more visible than the remaining sections tainted in gray in the navigation waveform, the black background is handy for public presentations on projectors;

• **Value** - As explained above, waveform colors are ordered by shades of gray to augment their respective saliency;

• **Texture** - The graphical objects of our waveform viewer are defined by plain RGB colors, they don’t make use of special textures;

• **Size** - The playback and navigation views share the same width so as to optimize the layout of the visual information for the same view angle sweeps of the user.

3.2.2 Visualizing temporal data

3.2.2.1 A short history of timelines

Grafton, A., & Rosenberg, D. (2010) provide an in-depth overview of visualizations and maps of events with a strong emphasis on time in the graphical representation. In short, the following items summarize the history of timelines:

- The earliest forms of timelines were time tables, with time and events as columns, providing a textual matrix-like representation.

- Throughout the 17th and 18th centuries, the advent of cartography mainly by Mercator, gradually shifted timelines towards a better use of the 2D space provided by sheets of papers. These time maps make use of a visual vocabulary “to clearly communicate the uniformity, directionality, and irreversibility of historical time”.

- From the mid-nineteenth, the advent of chronography lead to possibly obtain objective representations of timed events through recording, less biased by human interpretation. Marey and Muybridge invented cameras initially for observing natural phenomena. The science of metrology would allow to analyze really small and really large scales.

- Since then, visualization artists and scientists seized such recording tools and crunched massive datasets to propose a visual understanding of time-based events.

- More recently, Powerpoints and Prezis have been aiding a wider range of people to visualize, sequence and present information, in a multimedia experience, so as to support dialogues of expressing ideas. With the rising access of multimedia databases, complementary tools are appearing, allowing to create pathways in information. Recent projects such as Timeline from Vérité.co/Northwestern University and ChronoZoom from Microsoft/Berkeley allow users to quickly craft timelines on open topics using data from online repositories of media data such as YouTube, Vimeo, SoundCloud, Flickr...

3.2.2 Design cues for temporal data visualization

A design space composed of three tasks to be performed was proposed as framework for the visualization of temporal data by Aigner et al. (2007): modeling of the time domain and time-oriented data, computational analysis of the time-oriented data, and interactive visualization of the data.

Animations between visualization techniques switched during the task may arouse cognitive effects and improve the user’s comprehension of the underlying information present within the displayed data, as explored by Heer et al. (2007).
3.2.3 Visualizing media content

We follow up on this overview on temporal data visualization with specificities to some media types we chose to investigate: audio and video. We start by defining \textit{intra-} and \textit{inter-media navigation}, what helps to categorize media visualization techniques, in addition to the media type.

3.2.3.1 A foreword on inter-media and intra-media navigation

By \textit{inter-media navigation} we will denote the task of presenting relations between media elements of a database. By \textit{intra-media navigation} we will consider tasks consisting in examining the contents and information within a single media element.

The boundary between inter- and intra-media navigation can be set at the level of files. However, segments of a given media, once considered as separate media elements, can be visualized and organized by similarity in a browser (comparing shots, sequences, or even frames of a video; or choruses, verses, measures of music).

At the beginning of section 3.1 on organizing media content by similarity, we proposed a diagram describing a typical content-based dataflow (3.1). Let’s reuse the same terms that defined blocks of the dataflow, here in italics. Intra-media visualization essentially consists in producing visual \textit{thumbnails}, these are very specific to the media type (in our case audio or video), and not necessarily content-based. Inter-media visualization often relies on \textit{positioning} media nodes (each of which can emphasized with a thumbnail representation), the techniques for generating an inter-media representation are more generic, less media-dependent: when the organization is content-based, the most direct visual representation can be obtained by plotting the results of \textit{dimension reduction} applied to the feature set.

Based on this interpretation, we chose to proceed with media-specific overviews on intra-media visualization, respectively for audio in Section 3.2.3.2 and video in Section 3.2.3.3, then a media-independent overview on inter-media visualization in Section 3.2.3.4.

3.2.3.2 Intra-media audio visualization

Skilled musicians are able to recognize sudden changes in tempo, sense level dynamics (from \textit{piano} to \textit{forte}) and locate precisely the musical passage being played, from movements to bars and single notes; especially when they have studied beforehand the corresponding partition, that is a visual representation of the criteria necessary to reproduce the musical piece close enough to the way the composer meant it. The standard listener is less likely to be ear-trained so as to evaluate accurately the modification of parameters relevant to navigation such as playback speed, position or current sample, and amplitude.
3.2.3.2.1 Baseline representation: audio waveforms

The waveform is a traditional way of plotting a temporal signal. If we consider the etymology of the noun “waveform”, a visual representation (“form”) of a physical signal (“wave”), we understand that it aims at giving a visual feedback through an overall yet precise view of the general shape or form of a parameter evolving over time, commonly the amplitude in the case of audio signals, or often the envelope 27. From the signal plot many characteristics can be extracted, from points of interest by peak-picking to slopes by tangent estimation. Figure 3.4 captured from DJ software Mixxx by Andersen et al 28 displays two ways of using waveforms combined into their AudioFish view: a timeline that provides a visual summary of the whole file being analyzed (bottom layer); a zoom over the playback region that affords a fine-tuned navigation (top layer).

A survey of waveforms visualization techniques is proposed in 29, using visual variables to display more information than envelope or amplitude, rather: segments, frequency and timbral content, etc... Some advice is offered on how to visualize waveforms under small scale constraints, particularly by neglecting the negative part of the signal or subtracting it to the positive part so as to overlap both, similar to a half-rectified signal.

A regressive variation on these “mirrored graphs” called “n-band horizon graphs” 30, effectively reducing the height of time-series while keeping readability of information at high zoom factors, seems particularly useful for multitrack timelines.


More advanced audio thumbnails

Waveforms only allow to visualize temporal variations. This basic time-based representation of the amplitude could be augmented by considering the spectral, perceptual or semantic content of the signal. Gohlke et al.\(^3\) have been experimenting ways to map content-based parameters (for instance loudness and pitch) to colors. Mappings that seem to be perceived as natural are depicting loudness by color saturation and pitch by color brightness.

Bargraph-based visualizations use color to label comparative segments of audio tracks, that can be concatenated rectangular stripes as in Moodbar\(^3\), basically an augmented slider; or that are mapped onto waveforms as in the FindSounds.com Palette\(^3\); or even that are applied on “piano-roll” scores such as the “bar-graph score” of Music Animation Machine\(^3\).

There are also representations specific for musical signals. For instance, ImproViz by Snydal et al.\(^3\) aims at enlightening “signature patterns of a jazz musician’s improvisational style” through visualization called “melodic landscapes” and “harmonic palettes”. Isochords by Bergstrom et al.\(^6\), based on the Tonnetz structure, “make chords, harmonies, and intervals more salient to the viewer”, and are an example among others of music structure visualization based on geometric and mathematical properties (for instance topology and set theory). Arc Diagrams\(^7\) link similar song sections in “Shapes of Songs” by analysis of patterns in MIDI files.

Direct and primary visualizations that aid scientists in validating steps of the content-based dataflow are colorized 2D plots commonly used for the analysis of temporal signals, with frequency over time such as: spectrograms representing the audio spectrum, chronograms representing the melodic content (by sizing frequency bins to discretize these to desired note intervals) and rhythmograms emphasizing beats at different frequencies.

---

\(^3\) Kristian Gohlke, Michael Hlatky, Sebastian Heise, David Black, and Jörn Loviscach. “Track Displays in DAW Software: Beyond Waveform Views”. In: Audio Engineering Society Convention 128. 2010


\(^3\) Stephen V. Rice. “Frequency-Based Coloring of the Waveform Display to Facilitate Audio Editing and Retrieval”. In: 119th Convention of the Audio Engineering Society. New York, New York USA, 2005


3.2.3.3 Intra-media video visualization

Experts manipulating and editing video content need frame-accurate navigation and are accustomed to timeline-based non-linear video editors. While desktop video players have traditionally been only featuring a slider below the playback pane for temporal navigation, online video providers such as YouTube have recently augmented their video player with a keyframe visualization to improve the navigation. What are the alternate visual representations that could improve video navigation?

3.2.3.3.1 Baseline representation: video keyframes

Video content is often represented by its frames or keyframes in various ways:

- all frames aligned in time horizontally in a row of keyframes;
- a subset of these sequenced in time and overlapped in location (such as animated GIF image files serving as thumbnails on video hosting portals such as Archive.org);
- all frames are displayed on the same location, overlapped in time, as in a video player.

3.2.3.3.2 More advanced video thumbnails

Key information is sometimes hidden between keyframes.

Recently, face detection algorithms have emerged in video editing software such as Apple Final Cut X (to recognize relatives in rushes of family videos) or social media networks such as Facebook (for recommending tags on people). But what about the state-of-the-art among research works not yet accessible to a wide audience?

A recent survey covers most techniques for video visualization, considering spatial (images, video) and temporal media (video). Here follows an overview of a selection of video summarization techniques, some of them are borrowed from the survey.

“Slit-scans” are borrowed from analog photography and consist in concatenating a column (or row) of pixels of each frame of a video into a single image. This straightforward technique is particularly suited to videos featuring movement of recurring elements in the scene. “MotionGrams” by Jensenius et al. apply slit-scanning to the study of musical gestures. Tang et al. updated the concept of slit-scans into “slit-tears” by allowing users to annotate regions of interest through a pen-based system.
Chiu et al. proposed a “stained-glass” visualization where regions of interest are laid out in non-square fragments 42.

Ramos et al. 43 created a “twist-lens” technique where space occupation of an horizontal timeline is optimized by twisting the keyframe layout near the played back frame.

“Tapestries” by Barnes et al. 44 are inspired by medieval tapestries so that the temporal action of characters is represented spatially. A tapestry of Blender movie Elephants Dream is aligned in time with its “slit-scan”, both are are illustrated in Figure 3.5. Both techniques convey a similar overview in terms of the colors and shots present in the film. Tapestries focus on identifying movie characters and localizing their faces in time, whereas slit-scans provide a distorted but more cost-effective summary (just the time required to parse the movie frame by frame and extract a column of pixels from each).

Unfortunately commercial and professional applications for video editing, sequencing, collection management, etc... still make use mainly of video keyframes. This is probably due of a lack of robustness of feature extraction and segmentation techniques, the time taken to compute them in addition to the final representation, that can be interactive and thus require even more computational time and processing power.

42 Patrick Chiu, Andreas Girgensohn, and Qiong Liu. “Stained-Glass Visualization for Highly Condensed Video Summaries”. In: Proc. of the IEEE Intl. Conf. on Multimedia and Expo. ICME. 2004

Figure 3.5: Different techniques of visual summaries applied to Blender movie Elephants Dreams: slit-scan (top, computed by us) and “tapestry” (bottom, taken from Barnes et al.), both aligned in time.
3.2.3.4  **Inter-media visualization (by similarity)**

In his reference book on information visualization 45, Colin Ware reminds us that the works of the Gestalt school of psychology in 1912 can be considered as the first milestone in the field of pattern perception ("Gestalt" meaning "pattern" in German). Similarity being one of the eight Gestalt laws, Colin Ware recommends the use of "separable dimensions" or variables (for example color and texture) to visualize similarity.

Table 3.1 summarizes which inter-media visualization techniques have been employed in the browsers studied as part of our state-of-the-art chapter, among: a grid layout (□□□□□), a tree visualization (□□□□□), a scatter plot (□□□□□), or other advanced techniques (□□□□□). The context-based browsers we selected (notably tools for media experts) seem to favor grid and tree layouts, reminiscent of standard file browsers. The content-based tools we picked (especially research works) oscillate between grid and scatter layouts, with a tendency among intra-video browsers to use a grid layout since it optimizes space usage maximizing the amount of rectangular frames to be displayed. A few works we presented investigated cluster layouts (for instance connected graphs) and more advanced information visualization techniques (such as tree maps).

In Section 3.1.4, we mentioned some dimension reduction methods that aim at providing a low-dimensional representation of a feature set. When these dimensions are reduced to 2D or 3D, the results can be directly output in a scatter plot.

Other visualization techniques stem from the fields of machine learning, for instance dendrograms that display hierarchical clustering of the compared items through a 1D (sometimes 2D as in 46) representation with various sorting algorithms available to order the elements; and mathematics, particularly nearest-neighbor graphs and proximity drawings 47.

Simonin et al. studied the effect of 4 different layouts for browsing image collections 48: *Matrix-like* (a 2D grid); *Elliptic* (two concentric ellipses), *Radial* (a cross with eight radii, as in the *CrossBrowser*) and *Random* (scattered). 120 visual search tasks of each 30 images undertaken per 5 users were monitored through eye tracking. For each task a target image was displayed and the tester had to find it back and point it using a mouse. None of the layouts allowed a significantly faster target retrieval time, however the *Elliptic* layout proved to be the most comfortable visually in terms of reduced scan path length.

---


Very recently, Hudelist et al. have undertaken user experiments to compare different layouts of image browsers for mobile devices, with varying sizes of image collections. While their *Grid* view constitutes the baseline of the test as it is similar to default image browsers (images ordered in a matrix-like view), 3 other layouts sort images by color, in 3D views such as the *Globe* and *Ring* (similar to the authors’ video *3D Thumbnail Ring* presented earlier), and in the *Image Pane* 2D view. For the evaluation, each of the 48 participants was assigned to one layout among the 4 and had to find back a randomly-selected target image within one minute, this task repeated 60 times for each user and each collection (of 100, 200, 300 and 400 images). One important observation is that the average time to find the target grows somehow linearly for the grid layout along increasing sizes of collections, while the 3 other layouts tend to reach a plateau, faster for the 3D views. An ideal layout would be a morphing between the image pane at low collection sizes and one of the 3D views at higher sizes. Another remark obtained from the analysis that is worth considering when preparing the tests: some tasks were less successfully performed with the 3D views than with the other views, due to the fact that targets were located at the sides of the 3D objects, making these hardly visible when starting the task.

Both aforementioned experimental results underline that inter-media visualization (by similarity) is still an open research track. Other interactive information visualization techniques can be investigated to go beyond scatterplots and handle many dimensions, as surveyed in the state-of-the-art report by Kosara et al. This leads us to our next section focusing on interactivity.
3.3 Gestural input and control

We will start by providing general pointers about interaction through gestural input. Then, we will proceed with overviewing interaction devices often featured in media applications, uncovering how these lead to design interaction techniques (for media content navigation).

3.3.1 Generalities on gestural input

Interacting with computers in daily life is still governed nowadays by WIMP (window, icon, menu, pointing device) user interfaces, that are controlled by a mouse and a keyboard. The more recent rise of smartphones and tablets with touch input tends to alter this trend.

Gestural input theory has been studied for some decades. A notable work theorizing its design space is by Card et al. 51, gravitating around expressiveness (relation between input movements and output effects) and effectiveness (quantitative measures of performance), the latter including footprint (effects of form factor and size) and bandwidth (precision and scale of physical measures). Works more specifically tied to computer music applications have been gathered by Wanderley and Battier 52.

3.3.2 Gestural user interaction with media content

The first attempts of manipulation over recorded sounds were accidental, the most famous being Pierre Schaeffer’s looping of a sound passage in the late 1940’s episode of the “closed groove” caused by a scratch on the surface of a vinyl disc, becoming an audio modification technique among others. He would formalize and theorize the musical genre and practice called “musique concrète” that he initiated, also known as tape music, consisting in working directly on the medium where sounds are recorded so as to sequence events 53. Since then, disc jockeys have been developing and mastering advanced gestural techniques for controlling the playback of audio recordings. Traditional techniques for intra-media interaction follow mainly a time-based control paradigm, except for instance direct manipulation of scene objects in video content 54, 55.
<table>
<thead>
<tr>
<th>Browser</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Forager</td>
<td>1996</td>
</tr>
<tr>
<td>Raskin</td>
<td>2001</td>
</tr>
<tr>
<td>BumpTop</td>
<td>2006</td>
</tr>
<tr>
<td>Liquifile</td>
<td>2004</td>
</tr>
<tr>
<td>Film Finder</td>
<td>1993</td>
</tr>
<tr>
<td>AAU Video Browser</td>
<td>2010</td>
</tr>
<tr>
<td>Panopticon</td>
<td>2013</td>
</tr>
<tr>
<td>Sonic Browser</td>
<td>2001</td>
</tr>
<tr>
<td>Strands</td>
<td>2004</td>
</tr>
<tr>
<td>The amblr</td>
<td>2008</td>
</tr>
<tr>
<td>AudioFinder</td>
<td>1986</td>
</tr>
<tr>
<td>Soundminer HD</td>
<td>2002</td>
</tr>
<tr>
<td>Adobe Bridge</td>
<td>2005</td>
</tr>
<tr>
<td>Vintage Radio Interface</td>
<td>2012</td>
</tr>
<tr>
<td>Greenstone</td>
<td>2000</td>
</tr>
<tr>
<td>Bohemian Bookshelf</td>
<td>2011</td>
</tr>
<tr>
<td>SpeechSkimmer</td>
<td>1993</td>
</tr>
<tr>
<td>SoundFisher</td>
<td>1996</td>
</tr>
<tr>
<td>Marsyas3D</td>
<td>2001</td>
</tr>
<tr>
<td>SmartMusicKIOSK</td>
<td>2003</td>
</tr>
<tr>
<td>Musicream</td>
<td>2005</td>
</tr>
<tr>
<td>MusicMiner</td>
<td>2005</td>
</tr>
<tr>
<td>Search Inside The Music</td>
<td>2007</td>
</tr>
<tr>
<td>Music Rainbow</td>
<td>2006</td>
</tr>
<tr>
<td>Globe of Music</td>
<td>2007</td>
</tr>
<tr>
<td>MusicBox</td>
<td>2008</td>
</tr>
<tr>
<td>soniXplorer</td>
<td>2009</td>
</tr>
<tr>
<td>SoundTorch</td>
<td>2008</td>
</tr>
<tr>
<td>SongExplorer</td>
<td>2009</td>
</tr>
<tr>
<td>MuVis</td>
<td>2010</td>
</tr>
<tr>
<td>Sonarflow</td>
<td>2010</td>
</tr>
<tr>
<td>MusicGalaxy</td>
<td>2010</td>
</tr>
<tr>
<td>CataRT</td>
<td>2004</td>
</tr>
<tr>
<td>Informedia DLI</td>
<td>1995</td>
</tr>
<tr>
<td>FutureViewer</td>
<td>2006</td>
</tr>
<tr>
<td>VideoSOM</td>
<td>2006</td>
</tr>
<tr>
<td>ITEC Video Explorer</td>
<td>2010</td>
</tr>
<tr>
<td>MediaMill</td>
<td>2010</td>
</tr>
<tr>
<td>MediaTable</td>
<td>2010</td>
</tr>
<tr>
<td>Galaxy Browser</td>
<td>2011</td>
</tr>
<tr>
<td>Johaneum Browser</td>
<td>2012</td>
</tr>
<tr>
<td>OVIDIUS platform</td>
<td>2012</td>
</tr>
<tr>
<td>3D Thumbnail Ring</td>
<td>2012</td>
</tr>
<tr>
<td>Graphic Object Searcher</td>
<td>2012</td>
</tr>
<tr>
<td>HiStory</td>
<td>2012</td>
</tr>
<tr>
<td>3D Filmstrip</td>
<td>2013</td>
</tr>
<tr>
<td>Video Archive Explorer</td>
<td>2013</td>
</tr>
<tr>
<td>Sonic Mapper</td>
<td>2002</td>
</tr>
<tr>
<td>Pockets Full Of Memories</td>
<td>2001</td>
</tr>
<tr>
<td>Cell Tango</td>
<td>2006</td>
</tr>
<tr>
<td>Shape of Song</td>
<td>2001</td>
</tr>
<tr>
<td>Every Shot Every Episode</td>
<td>2001</td>
</tr>
<tr>
<td>cinematics</td>
<td>2011</td>
</tr>
</tbody>
</table>

Table 3.2: Browsers compared: interaction
3.3.2.1 Rotary control

Jog wheels have been widely used by experts of audio edition and video montage even before such systems were computerized. Wittenburg et al. \(^56\) ran an experiment to compare their system featuring rapid serial presentation visualization of video keyframes against traditional video playback as baseline system, playback speed adjusted using a jog wheel: their system was more accurate but not significantly faster than the baseline. Andersen \(^57\) studied the performance of jog wheels versus a click-based mouse interaction in musical segment search tasks, but found no significant difference in completion times.

Digital turntables use time-coded vinyls on which a synthesized sound is recorded, from which rhythm and playback direction can be easily extracted by signal processing, so that digital media can be played back using an analog metaphor \(^58\).

Snibbe et al proposed many haptic techniques for controlling media content \(^59\). For instance, their Video Caroussel, a 3D ring of TV channels, could be browser with the Sticky Channels mode that grant favorite channels a stronger force-feedback attraction. Beamish et al. \(^60\) experimented on how force-feedback would enhance audio navigation tasks. One of the elements that constitutes their D-Groove system is the Turntable Platter, a vinyl disc mounted on a motor. Several haptics effects were designed for it: bumps-for-beats makes the device bounce along the musical beats, resistance mode makes it harder to move in musical segments of high energy. IODisk by Tsukada and Kambara \(^61\) is a force-feedback rotary controller to browse digital content, particularly video and image collections. Also built with a disk mounted on a low-speed motor coupled with a rotary sensor, it allows to control the forward/backward speed of browsing in the collection and of playback for videos, including pausing when holding the disc still. Inter- and intra-media modes can be toggled using objects fitted with RFID tags are sensed by an RFID reader located on the device.

In our state-of-the-art chapter, Vintage Radio Interface and MusicRainbow featured rotary control for inter-media navigation, represented in column (β) of Table 3.2.
3.3.2.2 Tangible and touchable interfaces

For his PhD thesis, Ullmer designed tangible query interfaces to access and manipulate digital information, for instance the metaDESK allows to visualize and edit videos identified by tokens called mediaBlocks that act as “physical aliases”. But how does it scale up to huge collections of variable size? How does it allow to manipulate massive numbers of elements or fragments? A decade after, in a review of tangible user interfaces, Shaer and Hornecker still mention this scalability issue on top of the list of limitations and challenges of tangible user interfaces.

Hansen and Alonso studied how DJ techniques could be transposed to a multi-touch table (the reacTable). Such a device has been used by one of the browsers we analyzed in our state-of-the-art chapter: SongExplorer.

In our state-of-the-art table such a category is represented in column (3).

3.3.2.3 Free-form gestures and wearables

3D positioning with remote controllers such as the cheap Nintendo Wii has been employed in two content-based audio browsers we overviewed on our state-of-the-art chapter: SoundTorch and the amblr. Hayafuchi and Suzuki designed the MusicGlove for music manipulation by wavering fingers. Its media browsing abilities mimic what remote controllers offer: basic playback functionalities such as play, pause, next/previous track selection, volume modification. No user evaluation was performed with this prototype.

3.3.2.4 Audio input

Audio input, either performed by humans, or by recording sounds produced by other sources, can be an interaction technique to submit queries to browsing systems. For instance, query by beat-boxing, as proposed by Kapur et al., works by indicating rhythm and gross timbre mimicked by the mouth. Voice input with speech recognition could help produce instant user-defined tags.

---

3.4 Focus on rapid prototyping

In Section 3.1.5 we overviewed solutions to prototype applications and proofs of concept making use of multimedia information retrieval. However these solutions rarely provide the necessary tools for information visualization and interaction.

3.4.1 Scripted/textual versus visual programming

Signal processing and engineering specialists often use scripted and textual programming for their prototypes (for instance using Matlab by Mathworks) and they optionally switch to visual programming dataflow environments when realtime prototyping is of concern with tools like LabVIEW by National Instruments, Simulink by Mathworks. We believe that blending both approaches is convenient for the process of designing and prototyping the multimodal user interface of our adapted tool: visual programming gives a visual representation by itself of the underlying interaction pipeline, quite practical for exchanging design cues, while textual programming is quicker a designing simple and fast procedural loops.

3.4.2 Visual programming environments for interaction design

3.4.2.1 Visual programming tools for interactive arts

The number of multimodal prototyping tools and frameworks, dedicated to gestural input or generic towards most multimodal interfaces, has been increasing over the last two decades, yet none of them has been accepted so far as an industry standard. Anterior to these are data flow environments such as EyesWeb \(^{67}\), PureData \(^{68}\) and Max/MSP \(^{69}\) which often provide more usable visual programming development environments. We have been successfully using PureData as a platform for rapid prototyping of gestural interfaces. A notable feature from these environments that could be repurposed in the ones targeted for multimodal user interfaces: the “multi-fidelity” patch/pipeline representation modes of Cycling\textsuperscript{’}74 Max/MSP:

1. in “edit” or “patch” mode, the dataflow representation of the pipeline, widgets of processing blocks are editable and interconnections apparent between these;
2. in “running” or “normal” mode, widgets from the pipeline are interactive, but interconnections are hidden;
3. in “presentation” mode, widgets are “ideally” positioned as it would be expected from the prototyped user interface and connections are hidden as well.
3.4.2.2 Visual programming tools for multimodal interfaces

Among multimodal prototyping tools and frameworks, we would like to cite some that are still accessible and alleviate this issue of user-friendly prototyping. MaggLite by Huot et al. allows to sketch GUI elements and design the multimodal user interface by interconnecting blocks in a state-flow diagram. The Squidy Design Environment by Koenig et al. offers a zoomable user interface where details can be obtained from the user interface pipeline view by zooming smoothly onto nodes of the diagram, for instance to reveal more information on their connectivity, or to display temporal events more precisely on probe nodes. The OpenInterface engine with its Skemmi visual programming editor by Lawson et al. proposes two designer/developer modes switchable using a non-linear zoom slider.

3.4.2.3 Solutions for visualization

Regarding visualization, mostly libraries are offered rather than rapid prototyping tools. Preferred solutions are Prefuse in Java (that lead to many subsequent forks in different computer science languages: Flare in Flex, Protovis and D3.js in JavaScript) for information visualization; and VTK for the visualization of 3D computer aided design or medical content. VisTrails is one of the few projects that allows visual programming of workflows for data exploration and visualization.

3.4.3 Our choice

While several visual programming environments for prototyping multimodal user interfaces that have been overviewed above are still available, we opted for the PureData environment more generally suited for signal processing interests, also for visual prototyping, since it fulfills our expectations of creating simple pipelines for gestural input with basic devices and it is a well-established open source project.

3.5 Conclusion

In this chapter we exposed the method we want to employ to investigate further how to improve browsing tasks in audiovisual content, by combining: content-based organization by similarity, dedicated inter- and intra-media visualization, and gestural input for interaction.

More precisely, here are the tracks we opted to explore:

• Rather than developing new feature extraction and dimension reduction algorithms, we decided to proceed with state-of-the-art feature sets (available through the reusable YAAFE framework) and a new dimension reduction technique not yet widely used for sound and video collections: Student t-distributed Statistical Neighbor Embedding (tSNE).

• To focus on end-user aspects, we decided to leave out further investigation into complex segmentation algorithms. This leads us to concentrate our work on inter-media browsing on sound collections (involving short and homogenous sounds), and inter-media browsing on long video files (with simple visualization techniques).

• Rotary control has been widely used for intra-media navigation, less for inter-media browsing, sometimes augmented with interesting force-feedback cues. It provides a well-known metaphor borrowed from analog hardware for media browsing but it still requires validation of its performance against baseline solutions (using keyboards and mice) in the context of media navigation. Such reasons incline us to further study rotary control applied to media browsing.

• We opted for the PureData environment for the rapid visual prototyping of gestural input.
4 Design, prototypes, implementation

Baudelaire’s quote opens an issue. With the specialization of disciplines of (computer) science, new dedicated conferences have risen over the past decade, narrowing down the overall scope of the presentations to topics some scientists of other disciplines don’t consider scientific, for instance: New Interfaces for Musical Expression (NIME) and Tangible Embedded and Embodied Interaction (TEI) forked from the major international conference on Computer-Human Interaction (CHI). While not necessarily evaluated scientifically, the works presented there may still fascinate certain audiences: their potential and actual users, for instance participants to an interactive installation, creative people yearning for new interfaces for digital arts.

In this chapter we will present our prototyping environment, including the MediaCycle framework for prototyping content-based applications (4.1.2), the DeviceCycle toolbox for prototyping tangible interaction (4.1.4), both born under methodological constraints (4.1.1), supporting iterative short-term projects. This ecosystem enabled the design of several prototypes (4.2), we present a subset of them involving the author of the thesis.
4.1 Description of the prototyping environment

Our prototyping environment involved the creation of frameworks for prototyping content-based applications (4.1.2) and interaction (4.1.4), under methodological constraints (4.1.1).

4.1.1 numediart 3-month projects

This thesis was funded and time-framed by the numediart Research Program in Digital Art Technologies that ran between September 2007 and August 2013, through a grant from the Walloon Region of Belgium, towards Mons 2015 European Capital of Culture. Its founder Prof. Thierry Dutoit later turned the research program into the numediart Institute. One of the objectives of this program was to make state-of-the-art technologies accessible to artists and companies. Several research themes were defined, one being multimedia information retrieval. Collaborative projects between researchers, sometimes artists or industrials, were undertaken with a time frame of 3 months, expected to be concluded with working demos. How do sustained research and PhD theses fit in such an agenda? While it promotes regular writing activity, collaboration and the creation of prototypes on which research experiments can build upon, it needs to be associated with a longer-term vision, to daisy-chain projects on a sustained research theme. This working environment heavily influenced the pace of development of the prototyping environment this thesis builds upon.

Essays “From Gaia to HCI: On Multidisciplinary Design and Coadaptation” by Wendy E. Mackay and “Interaction Is the Future of Computing” by Michel Beaudouin-Lafon in 2 elaborate on multidisciplinary projects and engineering research, both aspects that had been central in running such short-term collaborative projects. How do engineers and artists come up with a common understanding, a common language? How do engineers from different disciplines (multimedia information retrieval from signal processing, and human-computer interaction from computer science) rank their objectives? Is a functional and improved algorithm enough as milestone, or is a user interface tested by users a necessary complement?

2 Thomas Erickson and David W. McDonald, eds. HCI Remixed: Reflections on Works That Have Influenced the HCI Community. The MIT Press, 2008. ISBN: 9780262050883
4.1.2 MediaCycle for the navigation by similarity

The MediaCycle framework allows to create applications for multimedia navigation and organization (by content-based similarity), making use of plugins dedicated for the support of different media types (audio, video, images, 3D models, text...) and for each sequence of the workflow of organization (file reading, feature extraction, thumbnailing, clustering or classification, computation of positions of media elements in a 2D space). As noted before, both this thesis and the development of MediaCycle has been interwoven in 3-month projects, as illustrated in Figure 4.1 clustering all MediaCycle-related projects by the media type each focused on. The framework itself has been developed in the long term in collaboration with colleagues from the numediart Institute (alphabetically): Stéphane Dupont, Alexis Moinet, Cécile Picard-Limpens, Thierry Ravet, Xavier Siebert, Damien Tardieu. Other colleagues participated sporadically to more specific projects.

![Image of media types]

Figure 4.1: MediaCycle-related numediart projects clustered by media type

4.1.2.1 Modular data flow

The data flow of MediaCycle has been designed to be sequential and modular, as illustrated in Figure 4.2. Sequential, since with MediaCycle only feature extraction is mandatory, provided that content-based organization is the primary objective of this framework, thus all subsequent phases can be bypassed. Modular, since each step of the data flow can be performed by a plugin.

In short, starting from a selection in a collection of media files, an offline process extracts features from each file, optionally segments these into smaller fragments giving more relevance to their comparison and assessment of similarity (a choice depending on temporal and spatial dimensions and homogeneity of content, based on features or heuristic methods), and produces thumbnails (that is one or more media files from each file, not necessarily of the same media type, aiming at summarizing the source media, again based on features or not, requiring segmentation or not).
From this point, if MediaCycle is integrated into an interactive application, GUI- or server/client-based, another process can be adapted on-demand: representations of each media file of the collection called media nodes can be clustered and positioned in low-dimensional spaces (for instance 2D) through dimension reduction for their display, rendering with optional thumbnails, visually and/or auditorily.

This data flow could be improved as follows:

- progressive sequentiality: allowing to display items progressively before these are fully imported, as an alternative to a progress bar, would blur the boundary between indexation and exploration times;

- media streams: beyond files that are already saved on a storage medium, the analysis of streams at recording time would allow content-based interaction modalities (for instance voice and sound through microphones, gestures through video cameras) and monitoring of live feeds.
4.1.2.2 Architecture

The architecture of MediaCycle has been progressively modularized and split into folders so as to allow various kinds of distributions, particularly for dealing with variable licenses among its components. Its file tree thus reflects its overall architecture, as illustrated in Figure 4.3.

The main application programming interface (API) is defined in its core library with minimal dependencies: core media support, classes for library and browser handling, management of the data flow through different plugin classes. The MediaCycle class holds access to all of these. Other folders hold a cascade of dependencies: 3rdparty contains third-party dependencies that are either tweaked or not available through OS-specific package managers; libs gathers media-independent libraries built upon the core library, mostly for (graphical) user interfaces; media contains media-specific libraries. All of these being dependencies of the plugin libraries contained in plugins.

Through a comprehensive integration of the CMake\(^3\) cross-platform building system for which supplementary scripts are available in cmake for packaging app in standalone bundles for OSX or packages for Ubuntu, all of these libraries and plugins can form media- or project-specific applications (in apps) and stripped-down applications for usability testing (in usability for the most fully-fledged, otherwise part of test). The last folder, doc, provides documentation to install properly the development environment and dependencies (using MacPorts\(^4\) for OSX and Personal Package Archives for Ubuntu \(^5\)), or to cross-compile (towards Windows from other operating systems with mxe\(^6\)), and a script to auto-generate the class documentation with doxygen which renders dependency graphs using graphviz.

---
\(^1\)http://www.cmake.org

\(^2\)http://www.macports.org

\(^3\)http://www.launchpad.net/~numediart

\(^4\)http://www.macports.org

\(^5\)http://www.launchpad.net/~numediart

\(^6\)http://www.mxe.cc
4.1.2.3 Media types

As a convention for shortened conversations, we would name MediaCycle-based applications by appending the suffix Cycle to another word describing projects or media types: for instance AudioCycle and VideoCycle are respectively the generic applications for browsing audio and video collections. The term MediaCycle was coined by Stéphane Dupont, initially as numediart project name for the second media browser then renamed ImageCycle. After AudioCycle, this new series of content-based applications raised the potential of converging towards a media-agnostic framework, followed by video, text, PDF and sensor, as illustrated in Figure 4.4. Names containing Archipel and Navimed refer to projects which required dedicated media types.

While classes derived from the parent class Media define the core properties of media types (for instance image/video width/height, audio sample rate, video frame rate...), Figure 4.5 illustrates how MediaData derived classes allow to abstract from third-party libraries that provide file input/output support. While the data containers describe and hold the storage representation of media content, mostly data types tied to input/output libraries, MediaData classes hold the media properties of the fragments and methods to access/open/close the data containers. This allows to keep access to data stored on the disk while being able not to keep the data in memory after importation. In short MediaContainer is at the level of the data type, MediaData at the level of accessing data (for now from files), and Media is an abstract view of the media content.
This three-level separation offers several advantages:

- generic libraries can be used as well as device-specific ones when the first can’t be compiled on specific targets, particularly mobile devices
- the size of the applications can be thus reduced, packing only the essential features

For instance, wherever the size of the dependency is not important and the dependency compiles against the architecture easily, a combination of video data container with respectively FFmpeg to support a large number of formats and OpenCV to access to numerous computer vision algorithms would make sense. In contrast, on tablets with lower storage capacities, different architectures than these on which the aforementioned libraries were developed, using the stripped-down libraries provided by the operating system would be the preferred solution.

http://www.opencv.org
http://www.ffmpeg.org
4.1.2.4 Plugins

While describing the data flow before in Section 4.1.2.1, we mentioned that all phases could be performed by plugins. Figure 4.6 illustrates the plugin class hierarchy refined up to the writing of this thesis. We will describe the most salient of these in the coming sections, to underline how their development took a significant time. The existence of a base plugin class justifies itself by the fact that all plugins need common methods to be opened, closed, reset, provide information (name, description, supported media type), informed of the current status of the data flow, and get an instance to the main MediaCycle class.

![Diagram showing the MediaCycle classes: base plugins](image-url)
4.1.2.4.1 Plugin parameters
We introduced the possibility to serialize parameters for any plugin class of MediaCycle, making each controllable by automatically-generated GUI palettes. As illustrated in Figure 4.7, these are of three types: callbacks, numbers and strings. For instance a scatter plot position plugin can lay out media elements on a 2D cartesian space, with time on the x-axis, and any feature on the y-axis. There the axis assignment parameters would both be string parameters, represented by pull-down menus in the GUI.

As illustrated in Figure 4.8, MediaCycle plugins are contained in plugin libraries that are either dynamically loaded (useful for lighter-weight remote updates of applications or to fit with different software licenses and the entailed marketing issues) or statically compiled (so as to reduce size and loading time). Default plugin libraries are part of static libraries linked against all applications to make sure that at least a basic data flow with all required plugins is allowed with any distribution.

4.1.2.4.2 Media library plugins
Media library plugins were integrated first and foremost for exporting a subset of the information produced by MediaCycle to other complementary applications (spreadsheets, numerical analysis), while having in mind possible and future compliance with other content-based platforms or standards, for import and export as seen in Figure 4.9.
4.1.2.4.3 Media reader plugins

Media reader plugins generate instances of classes inherited from Media, MediaData and MediaDataContainer. These are generally named after the libraries for file/media input that these use, as illustrated in Figure 4.10. They also return information on supported file types.

![Figure 4.10: MediaCycle classes: media reader plugins](image-url)
4.1.2.4.4 Media features plugins

Media features plugins allow to extract information out of media elements, from content-based features to simple file-based properties. As a mirror to the MediaDataContainer presented before, we considered three classes of feature plugins: temporal, spatial and without dimensions, however we did not implement so far use cases supporting spatial features. Figures 4.11 and 4.12 illustrate the available feature plugins: basic FeaturesPlugins allow to extract vectors of multiple dimensions, TimedFeaturesPlugins support matrices with one multi-dimensional feature vector per time frame.

![MediaCycle classes: media features plugins](Figure 4.11)
For audio content, we relied on some of the numerous features offered by the Yet Another Audio Feature Extraction (YAAFE) library. A wrapper plugin around the VAMP plugin SDK, a tentative standard for music information retrieval, is also available. For video (and image) content, all the features are computed with the renown OpenCV library: optical flow, color spaces, pixel speed and rigid transform. Other attempts with GPU acceleration (Nvidia CUDA) and bindings to the EyesWeb platform were also investigated through projects.

![Diagram of MediaCycle classes: timed-features plugins](image-url)
4.1.2.4.5 Media segmentation plugins Temporal segmentation has primarily been investigated, as opposed to spatial segmentation that would extract regions of interest or objects in visual media. We implemented two segmentation plugins based on the Bayesian Information Criterion: the first browses the data frame-by-frame while the second implements a "divide-and-conquer" approach\(^9\). The latter method is faster, but the former can be applied to real-time segmentation on a video stream. Another segmentation method, based on the self-similarity matrix to compute a signal of novelty\(^10\), has also been integrated. Due to time constraints and focus, we didn’t use segmentation for experiments in this thesis since these required manual parameter tweaking or the implementation of interactive machine learning to alleviate this.


Figure 4.13: MediaCycle classes: segmentation plugins

4.1.2.4.6 Media thumbnailing plugins In 2013, we added the support of media thumbnails, since the choices of mappings of content-based features to visual variables was restrictive and fixed: distance and similarity conveyed through the position of the media elements in the visual representation, and to some extent colors tied to clusters. What about the other visual variables, such as shape/contour, texture? Thus, our definition of a media thumbnail is generic: conveying summarized information on a given media element by computing one or more media element(s) not necessarily of the same media type.
Figure 4.14 lists all the thumbnailing plugins created so far. Their role is to produce a vector of MediaThumbnail instances for each supported Media element and to implement virtual methods querying minimal information about the thumbnails: name, description, media type, extension, and their location on the data flow: whether they require feature extraction and/or segmentation. For example, state-of-the-art thumbnails not requiring feature extraction nor segmentation are: an audio waveform as summary of an audio file, video keyframes evenly-sampled from a video file. Thumbnails can be static (pre-computed) or dynamic (adapted based on user interaction or playback synchronization).

4.1.2.4.7 Clustering plugins

Clustering is a subset of machine learning and classification, as it can be otherwise called automatic classification. It is mainly used for exploratory data analysis. One way to convey similarity in large collections of media files is to group them by clusters, or groups of elements whose features are minimally distant. This allows to speed up the exploration and search time, for instance by only previewing a few representatives of each cluster. If there is only one to preview per cluster, it is called the centroid, representative of all the other elements from the same cluster. The main method used throughout MediaCycle projects is the renown k-means, converging towards a user-defined amount of $k$ clusters. As illustrated in Figure 4.16, the MediaCycle team progressively made derivations out of it, notably to introduce the support of interactive tagging.
From the beginning of MediaCycle, the linear algebra library armadillo has been used for the implementation of these clustering plugins. More recently, a library build upon it, mlpack, was also introduced since it provides other optimized machine learning algorithms.

A wrapper to the Octave linear algebra framework was integrated into MediaCycle, initially to allow the classification of makam music (mostly from Turkey) sourced from Matlab scripts. Such a wrapper allows rapid testing and prototyping of results and algorithms from the music/multimedia information retrieval community, at the expense of slower code execution.

![Figure 4.15: MediaCycle classes: cluster method plugins](image-url)
4.1.2.4.8 Neighborhood plugins

The workflow of exploratory media navigation can be performed through different scenarios combining different phases. For instance, after having gained an overview of a quite large collection, the user may stumble onto one specific element and decide to discover similar elements. To do so we introduced an alternate mode located at the same stage as clustering in the data flow and that computes neighbors of a given element. As illustrated in Figure 4.16, the implemented methods mainly differ from the chosen distance metric or distribution: either an Euclidean distance, or a Pareto distribution, or a random order (the later useful mostly for testing positioning algorithms). While clustering method plugins are applied on a vector of media elements representing the whole collection, neighbor method plugins feed a tree of media elements with similar elements as children.

4.1.2.4.9 Positioning plugins and dimension reduction

When collections need to be displayed visually, traditionally on 2D screens, methods are required to scale the usually large feature space computed from the media files down to a few dimensions. A simple method consists in assigning one feature per axis in a scatter plot, but this implies that the features are unidimensional. Dimension scaling algorithms allow to bridge this gap, some succeeding better than other into preserving closest neighbors between the feature space and the visual space by attempting to display similar elements close one another.

As illustrated in Figure 4.17, we introduced several positioning plugins:

- the inaugural method nicknamed Propeller would clearly lay out elements in their distinctive clusters resulting from k-means clustering into a helix visualization;
- some are solely dimension reduction techniques, such as Principal Component Analysis (PCA), Multidimensional Scaling (MDS), and variants of Statistical Neighbors Embedding (SNE) adapted with probabilistic methods such as the Kullback Leibler (KL) divergence or the Student-t distribution (names containing t_);
- some are borrowed from information visualization, such as the cartesian scatter plot or the radial/polar ClustersClock and Gramophone;
Figure 4.17: MediaCycle classes: cluster position plugins
The neighbor method plugins described in paragraph 4.1.2.4.8 require specific position plugins that display a subset of the collection that is organized into a tree of similar elements. Figure 4.18 shows the two positioning plugins dedicated for neighborhoods representations. Both representations fork implementations from the Prefuse library. The straight node-link tree layout algorithm optimized for fast execution has been designed by Buchheim et al. 15. The radial node-link tree layout has been created by Yee et al. 16.

Figure 4.18: MediaCycle classes: neighbor positions plugins

4.1.2.4.10 Media rendering plugins
Rendering consists in displaying media content into channels or modalities that are perceivable by humans. As illustrated in Figure 4.19, two such channels can be exploited with MediaCycle: sight and audition.

A first group inheriting from OsgRendererPlugin shows visual renderer plugins per media type, implemented with OpenSceneGraph, an OpenGL scene graph manager on top of which the MediaCycle visual browser is built, a choice preceding the start of this thesis. If this choice was to be revisited today, candidates such as HTML5 would open to web-based applications; and Qt, already used to the standard MediaCycle GUI, would integrate the HTML5-based visual browser. These plugins render media elements at various states: default “still” visualization when interaction is inactive, when hovered by the user, when their playback is active or not, when metadata details are requested. The more recent media node feature mapper plugins attempted to provide media-agnostic methods to map information obtained from the features to visual variables: assigning color beyond cluster identifiers, adapting the contour of nodes along temporal variations of some features, and so on, with currently one-to-one dimensional mapping.

The remaining plugins are different variations of audio playback plugins, mainly for trying different sound rendering APIs such as OpenAL, PortAudio and the Sound Synthesis Toolkit (STK) as required through various projects: the first comes from the game industry and packs perceptual and spatial effects, the second offers a better latency, the third comes from the computer music research community and provides standard digital audio effects.
4.1.2.4.11 Client plugins

Client plugins allow remote control and navigation of MediaCycle collection, which is useful in two major cases: 1) a server indexes a large scale content and clients connect to it to browse a subset, 2) an application integrates the MediaCycle browser, but the interaction with it requires to be flexible or modular. Our concerns were mostly motivated by the second category.
Figure 4.20 illustrates various client plugins. For instance, MediaCycle-based applications can be controlled through the OpenSoundControl (OSC) protocol, using a network communication even if in most of the cases both applications are running locally. We also adapted our system to contest in the Video Browser Showdown, a competition-based usability evaluation requiring the submission of a target media element to a server, and needed to pursue similar known-item search usability tests locally, for which a dedicated controller allowed to submit targets.

At the beginning of this thesis, AudioCycle was a standalone project, with classes contained in one folder, only working for Apple OSX since its GUI was based on the Apple Cocoa SDK and its compilation environment restricted to the Xcode IDE. Ever since, we ported the GUI to the cross-platform Qt library to cover the major operating systems. We granularized the framework as it has been described in this section, up to the point that creating a new application consists simply in declaring a few options in a CMake script: name and description of the application, supported media type, list of required plugin libraries, optional default plugins for each type. We ended up with one application per media type and/or per numediart project, which the script-based system made easier to maintain against API changes. Some of these applications will be described in section 4.2.
4.1.3 OpenSoundControl (OSC) for remote control

One of the aforementioned client plugins allowed to control MediaCycle applications externally with any other application supporting the OpenSoundControl (OSC) protocol. Pre-formatted messages to be sent through a network (that can be hosted on the same computer) from the interaction prototyping application to the MediaCycle application that will interpret these messages. As it is now done in many applications requiring gestural input, we have defined our own OSC namespace, a taxonomy of messages, to control the AudioCycle application.

Initially, MediaCycle supported interaction with a single user but the namespace was designed open to collaborative and concurrent use that was later implemented.

4.1.4 DeviceCycle, a PureData toolbox for gestural input

PureData is a graphical modular environment for data flow processing that has been developed nearly since three decades, initially for manipulating audio and control signals such as the Music Instrument Digital Interface (MIDI) protocol, more recently processing video as well, and supporting the Open Sound Control (OSC) network communication protocol. It resembles Max/MSP by Cycling’74 for a simple reason: both originate from the same ancestry, born at Institut pour la Recherche et Création en Acoustique et Musique (IRCAM) in Paris, France to satisfy interactive computer music requirements. Data flow schemes named patches are produced by connecting signal blocks called objects by their inlets and outlets in a canvas. External objects can be easily added through various APIs, either the native in C language for each environment (PureData and Max/MSP), or using flext in C++ that supports both.

This solution for the rapid prototyping of user interfaces was adapted here because: it is free and open source, cross-platform, it precedes multimodal interface design applications (signal processing being a method to achieve so), and while complex multimodal fusion wasn’t required for this thesis, PureData offers lots of signal filtering and mapping objects to back up most scenarios.

---


24 Thomas Grill. “flext - C++ layer for Pure data and Max/MSP externals”. In: Linux Audio Conference. 2004
We have created patches for the following devices:

- USB Human Interface Devices (HID), particularly Contour Design Shuttle jog wheels (Figure 4.22) and 3dconnexion Space Navigator 3D mice (Figure 4.21), using the [hidio] object by Steiner et al \(^\text{25}\), offering several improvements over [hid] (notably hotplugging devices);

- the Novint Falcon 3DOF force-feedback mouse (Figure 4.23), using the [np_nifalcon] object using a reverse-engineering driver library\(^\text{26}\), assorted with the HSP set of abstractions by Berdahl et al

- the Apple Macbook Multitouch Trackpad, allowing to access information of blobs detected from fingers touching its surface, using the [fingerpinger] object that we ported from its initial Max/MSP implementation\(^\text{27}\) to PureData, only available on Apple OSX (Figure 4.24).

Since some of the patches mimic the layout of the devices themselves in some kind of degraded minimal realism, mappings can be changed quickly on the go and fine-tuned without recompiling. We released this toolbox for free hoping it may help other people spare some time during their prototyping tasks. We made this toolbox available on github \(^\text{28}\) under a GPL license.


\(^{26}\) http://sf.net/p/libnifalcon

\(^{27}\) https://code.google.com/p/anyma-max-externals/

\(^{28}\) http://github.com/ChristianFrisson/DeviceCycle
Figure 4.22: DeviceCycle patch for the Contour Design Shuttle Pro2 jog wheel (left) and picture in use (right).

Figure 4.23: DeviceCycle patch for the Novint Falcon 3DOF force-feedback “mouse” (left) and picture in use (right).
For our rapid prototyping concerns, we developed patches based on this toolbox to control MediaCycle-based applications via the Open Sound Protocol (OSC) protocol. With this approach, we produced several test prototypes that were first presented in 2010 for the control of the AudioCycle application:

- a keyboard and multitouch trackpad combination
- bimanual control with a 3D mouse for intra-media navigation (between elements of the collection) and a jog wheel for intra-media manipulation (inside one element, temporally);
- a force-feedback version for the navigation, replacing the 3D mouse by a 3DOF force-feedback mouse;

Brent offers an alternative named Dlib also developed with PureData that appeared right after DeviceCycle, packaging alternative devices such as the Nintendo Wii remote controller and the Kinect depth-sensing camera.

---


31 http://williambrent.conflations.com
4.2 Designs and prototypes

Several interactive applications have been created using the MediaCycle framework, covering diverse users, different media types, dedicated interaction means. Figure 4.25 timelines numediar projects around media browsing that had either a relation to this thesis (4) and/or made use of the MediaCycle framework (4). Past the end of the numediar funding (March 2013), projects are allocated to the closest quarter.

<table>
<thead>
<tr>
<th>January-March</th>
<th>April-June</th>
<th>July-September</th>
<th>October-December</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio Skimming</td>
<td>Audio Thumb nailing</td>
<td>Video Navigation Tool</td>
<td>Audio Cycle</td>
</tr>
<tr>
<td>Image Cycle</td>
<td>AV Laughter Cycle</td>
<td>CoMedi Annotate</td>
<td>Multimedia Cycle</td>
</tr>
<tr>
<td>MoVi</td>
<td>Audio Garden</td>
<td>Media Blender</td>
<td></td>
</tr>
<tr>
<td>Media Blender</td>
<td>LoopJam</td>
<td>RealTime MediaCycle</td>
<td></td>
</tr>
<tr>
<td>DiskHover</td>
<td>Media neum</td>
<td>Makam Cycle</td>
<td>VideoBrowser Showdown</td>
</tr>
<tr>
<td>Mashta Cycle</td>
<td>Mashta Cycle</td>
<td>The Listening Room</td>
<td>LavaAMP</td>
</tr>
<tr>
<td>Tangible Needle</td>
<td>Audio Metro</td>
<td>Auracle</td>
<td></td>
</tr>
</tbody>
</table>

A subset of these projects (4) will be analyzed in thereafter, focusing on projects that produced results of interest for this thesis, particularly media browsers. For each browser, the motivation and intention of their creation will be clarified, their design and prototyping will be summarized while citing scientific papers by the authors of this thesis providing further details, and the contributions of the author of this thesis will be underlined.
4.2.1 AudioCycle: manipulating audio loops (2008)

Figure 4.26: Screenshot of the initial AudioCycle (2008)

media granularity organization visualization interaction users setting desk community availability / usability

Table 4.1: Taxonomy

4.2.1.1 Motivation and intention

AudioCycle results from the first numediart project among a MediaCycle series that aimed at reproducing the state of the art of inter-media browsing of media content, to discover the field of multimedia information retrieval first with audio content.

4.2.1.2 Design and prototype

To benchmark each phase of the multimedia information dataflow (that we illustrated in Figure 4.2), researchers usually rely on metrics. Before choosing the proper metric, scientific visualization may aid to generate a visual overview of the results. The first common visualization technique tried is a scatter plot, with features assigned per axis, or a mixture of them reduced to 2D.

The initial goal of AudioCycle was hence to explore visualization techniques of media collections following the analogy of a galaxy. Before the current thesis started, colleagues came up with a visualization technique named Propeller that would group elements into a user-defined amount of clusters. Starting from a central “query” element, centroids of each clusters are evenly-distributed angularly around the center, and all elements from each cluster are distributed from their high-dimensional distance to both the center element and their centroid. This initial AudioCycle prototype has been presented at two IEEE-sponsored conferences in 2009.33,34.

4.2.1.3 Contributions

It is during the creation of AudioCycle that the need of prototyping gestural user interfaces beyond mice and keyboards was raised by the author of this thesis. The DeviceCycle toolbox for PureData using the OpenSoundControl protocol to communicate with MediaCycle applications, described earlier in this chapter, had been conceived at that time for that purpose.

A straightforward bimanual demo binding a 3Dconnexion Space Navigator 3D mouse for inter-media browsing and Contour Design Shuttle Pro2 jog wheel for intra-media browsing was set up as example (in Figure 4.27) and presented as poster at the 2010 edition of the New Interfaces for Musical Expression (NIME) conference.35 Further user studies are necessary to (in)validate the benefits of designs such as this one.

AudioCycle was initially implemented with the Cocoa framework for GUI applications running only on Mac OSX platforms. Porting it to the cross-platform Qt framework, was one technological contribution by the author of this thesis, thus extending perspectives for potential collaborations.
4.2.2 AudioGarden: morphing rhythmic/timbral sounds (2010)

![AudioGarden screenshot with the Gramophone layout](image)

Motivation and intention

AudioGarden proposes a new approach to sound composition for soundtrack composers and sound designers.

Cécile Picard-Limpens’ doctorate studies focused notably on providing solutions for the realtime creation of generative soundtracks from audio content for video games. Met at the eNTERFACE’09 Summer Workshop on Multimodal User Interfaces in Genova in 2009, discussions lead to a shared understanding that her methods developed within Matlab prototyping environment would be more accessible outside digital signal processing labs if these were implemented in a standalone application including a browser for sounds. A Short-term Scientific Mission (STSM) was awarded to her by European Cooperation in Science and Technology framework COST Action IC0601 Sonic Interaction Design (SID) to undertake visiting research at the numediart labs and further develop this idea with the numediart team working on AudioCycle.

---

36 Cécile Picard, Nicolas Tsingos, and François Faure. “Retargetting Example Sounds to Interactive Physics-Driven Animations”. In: AES 35th International Conference on Audio for Games. 2009

4.2.2.2 Design and prototype

We first automatically segment audio recordings into atomic grains which are displayed on our navigation tool according to their timbre. Two visualization techniques were designed for that purpose. The *Gramophone* (in Figure 4.28), a polar layout with a timbral feature on angle (first principal component of MFCCs) and inverse duration on radius, is useful for sound exploration and grain/pattern discrimination. The *Flower view* (in Figure 4.29), with sounds on a scatter plot formed by timbral features, with sound segments circularly distributed around their parent recording, helps to trace timbral origins.

To perform synthesis, the user selects one recording as model for rhythmic pattern and timbre evolution, and a set of audio grains. Our synthesis system then processes the chosen sound material to create new sound events based on onset detection of the recording model and similarity measurements between the model and the selected grains.

4.2.2.3 Contributions

We presented *AudioGarden* as poster and side demo at the Audio Mostly conference in 2010. The author of this thesis allowed the control of synthesis parameters through panels and synthesis actions through shortcuts.

---

4.2.3 LoopJam: playing a collaborative musical map (2011)

4.2.3.1 Motivation and intention

Damien Tardieu, member of the MediaCycle team at the time, coined the idea that hooking up a depth sensing camera to the AudioCycle application would result in a demo for exploring sounds on a map by navigating physically over the space, targeted for novice users, beyond multimedia information retrieval specialists and sound experts 39.

4.2.3.2 Design and prototype

The LoopJam installation allows multiple visitors to compose instant musical creations, with a free-form interface sensing their center of gravity, corresponding to a pointer incidentally triggering the closest loop in a 2D map of sounds.


40 https://code.google.com/p/qtkinectwrapper/
of their center of gravity through the OpenSoundControl protocol to a stripped-down version of AudioCycle (its browser display fullscreen without panels). Each participant’s position is mapped to a pointer triggering one sound that is synchronized with the others by phase vocoding for pitch-independent variable-speed playback. The installation cycles through sound collections similarly to a jukebox or music videos broadcasts.

4.2.3.3 Contributions

The author of this thesis implemented the multi-pointer support in MediaCycle allowing to assign one pointer per person. LoopJam has been presented at various artistic and scientific venues:

- the Arts & Sciences exhibition at the Seneffe Estate, Belgium, May 21-22 2011
- Centre Wallonie-Bruxelles (CWB) in Paris, France, Sept. 22 - Oct. 23 2011
- the 2011 Networked & Electronic Media (NEM) Summit in Torino, Italy, Sept. 27-29 2011
- Interaktive eKsperience, Kikk international festival, Namur, Belgium, Nov 8-9 2012
- Maison Folie, Mons, Belgique, for the Transnumériques #4 festival from Transcultures and les Journées d’Informatique Musicale (JIM), May 9-11 2012
- New Interfaces for Musical Expression (NIME), Ann Arbor, Michigan, USA, May 21-23 2012
- Underground hall of movie theater Galeries for the Citysonic #10 festival, Brussels, Belgium, September 27 to October 14 2012
- Musical Instruments Museum (MIM) in Brussels, Belgium, through the annual week of La Semaine du Son, January 22-27 2013

---

4.2.4 MashtaCycle: mashing-up sounds through gestures (2013)

4.2.4.1 Motivation and intention

Belgian artist Gauthier Keyaerts answered a call for creation of sound collections for LoopJam for its installation during the Citysonic 2012 festival. Gradual discussions with the author of this thesis motivated both to iterate over the design of LoopJam beyond just triggering sounds by hovering the space, towards a musical instrument with extended gestural control and more impact on the sound rendering.

4.2.4.2 Design and prototype

An artistic visual rendering controlled by gestures and sound was also envisioned by the artist. The relevance of visual feedback of the sound browser was questioned by aesthetic and hardware constraints: monitoring the sound collection would smear on the visuals and require additional display devices. Sound collections featured complex sounds that were quite dissimilar. Content-based similarity was put into jeopardy.
Fabien Grisard visited the numediart Institute between March-August 2013 for his Masters Thesis\(^44\). He integrated new gestures into Felix Kistler’s Full Body Interaction Framework (FUBI) \(^45\) and added an OpenSoundControl bridge to control AudioCycle through free-form gestures, some inspired by Sound Painting. Mappings were first prototyped with PureData (as seen in Figure 4.32) then hardcoded in C++.

A new audio engine with effects (reverb, buffer freezing) was developed based on the Sound Synthesis Toolkit (STK) \(^46\). Since this fast-prototyped audio engine was not stable enough and content-based similarity felt not required, MashtaCycle was terminated to lead to a different project without the author of this thesis: Fragments #43-44 by Gauthier Keyaerts.

4.2.4.3 Contributions

The author of this thesis called for and supervised Fabien Grisard’s Master thesis and created the new audio engine. MashtaCycle was presented at:

- the Citysonic festival in Mons, Belgium, September 6-22 2013
- the INTETAIN conference in Mons, Belgium, July 3-5 2013 \(^47\)


\(^47\) Christian Frisson, Gauthier Keyaerts, Fabien Grisard, Stéphane Dupont, Thierry Ravet, François Zajéga, Laura Colmenera Guerra, Todor Todoroff, and Thierry Dutoit. “MashtaCycle: on-stage improvised audio collage by content-based similarity and gesture recognition”. In: *Proceedings of the 5th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN)*. Mons, Belgium, 2013. DOI: 10.1897/978-3-319-03892-6_14
4.2.5 The Listening Room: exhibiting sounds from paintings (2013)

4.2.5.1 Motivation and intention

Rudi Giot and Carla Scaletti, organizers of the 2013 edition of the Kyma International Sound Symposium (KISS) in Brussels, invited us to install LoopJam in the foyer of the host venue, the ISIB Engineering School of Brussels, to display sound collections created with the Symbolic Sound Kyma sound design system and each inspired by a painting of René Magritte, theme of the 2013 edition.

Named after Magritte’s painting The Listening Room (1957), the installation was actually forked from MashtaCycle to benefit from extended gestural control over sound.
4.2.5.2 Design and prototype

The Listening Room made use of two projectors: one projecting on a wall the browser of sounds as in LoopJam; and also another displaying the visual rendering of the depth sensing camera on the ground, since gestures would only be detected if bodies were properly tracked and segmented, what required some visual feedback.

4.2.5.3 Contributions

The Listening Room was presented at the KISS 2013 on September 12-15 2013 in Brussels. It featured collections from: Ilker Isikyakar (USA), Gauthier Keyaerts (BE), Olga Oseth (USA), Carla Scaletti (USA).
4.2.6 VideoCycle: mining passages in long videos (2013)

4.2.6.1 Motivation and intention

We decided to compete with VideoCycle in the Video Browser Showdown at the Multimedia Modeling 2013 conference in Huangshan, China, against other video browsers through Known-Item Search (KIS) tasks in long videos, so as to get acquainted with this evaluation method and to get inspired by other browsing tools.

4.2.6.2 Design and prototype

While the DeviceCycle toolbox would allow to fast prototype input modalities for user interfaces, it wouldn’t fit with the constraints of a live competition: launching a single standalone application would be safer and faster. We thus implemented the external HID device control with the hidapi library directly as MediaCycle plugins bundled with the application.

Initially, a fully-fledged version of VideoCycle was prepared for the contest. As pictured in Figure 4.34 (where yellow dotted lines/arrows are annotations), users would zoom and pan the inter-segment “browser” (up) with a multitouch trackpad, and scrub the intra-media “timeline” (down) with a jog wheel.

---


49 http://www.signal11.us/oss/hidapi/
However the laptop brought to the competition couldn’t hold hundreds of video players corresponding to segments of long recordings in the browser view, while the desktop-based development computer could. Thus a minimalistic version of VideoCycle featuring only its timeline was used during the contest, without content-based organization abilities, purely relying on its interaction and visualization techniques.

4.2.6.3 Contributions

The evaluation of the 2012 Video Browser Showdown session was published in the International Journal of Multimedia Information Retrieval in December 2013. Observations restricted to the performance of VideoCycle during the competition are summarized in next chapter.

The author of this thesis implemented the video timeline and HID device plugins.

---

4.2.7 **LavaAMP: streaming colorful blobs surrounding music (2013)**

4.2.7.1 **Motivation and intention**

The author of this thesis and Eric Schayes, a student colleague from the ISIB Engineering School of Brussels, answered the call for participation to the fifth annual Student Innovation Contest at the 26th ACM Symposium on User Interface Software and Technology (UIST) in 2013. For this contest, students had to design within a few months creative applications dedicated to the specific hardware of the year. In 2013, the Microsoft PumpSpark Fountain Development Kit, 8 water pumps controlled by a board through the USB port was selected.

Our prototype, LavaAMP, surrounds the beats in music tracks by streaming colorful blobs. *LavaAMP* is inspired from vacuum tube amps and lava lamps.

4.2.7.2 **Design and prototype**

*LavaAMP* was housed in a refurbished vinyl player enclosure. 3 plastic bottles were fitted each with a pump immersed into 90% of edible organic oil (here from colza for great clarity), 9% of water diluting 1% of ink of unique color per bottle, backlit.

The PumpSpark board including a cooling fan was positioned behind a side grill of the vinyl enclosure where loudspeakers were initially located.
The whole prototype was implemented in the PureData prototyping environment: pump actioning using the [comport] object and audio analysis using aubio objects for beat tracking. Each onset detected in music tracks would trigger an pump impulse in one bottle with cyclic permuting order, creating a flow of colored blobs. Initially we wanted to perform onset detection making use of the perceptual Bark frequency scale using Brent’s timbreID objects for PureData, assigning 1 spectral band per bottle, but it happened to be less visually striking, not necessarily emphasizing salient beats.

4.2.7.3 Contributions

Eric Schayes bottled the pumps and hacked the PureData patch to actionate them. The author of this thesis came up with the furniture design and hacked the audio analysis.

Getting access granted to this Microsoft PumpSpark Fountain Development Kit also benefited other students from the University of Mons and ISIB Engineering School. They would challenge with their designs to other various national contests.


http://aubio.org

4.2.8 Grayfish Squidget: tangible intra-media browsing (2014)

User interface design cycles can be imprinted in feedback loops, notably when film makers inspire with their creative dreams repurposing technologies as featured in science fiction movies, while technologists end up designing user interfaces tailored for media content organization, manipulation, serving media arts. For instance, in movie The Final Cut (Naim, 2004), the protagonist uses a steampunk-like desk with a wooden interface to edit rememories, movie montages from memory implants of bygone people. How could we reproduce a familiar yet inviting setup for media navigation? Through tangible needles, in other words, tangible user interfaces inspired by past technologies related to media practices?

4.2.8.1 Motivation and intention

For intra-media navigation, the grayfish squidget, a refurbished portable vinyl player has its inside mechanics replaced by a webcam monitoring circular gray code analyzed through computer vision for position/speed tracking.
A circular Gray code disc printed using Wheel Encoder Generator has been fitted to both inner and outer faces of the turntable. Inside, for speed and position tracking including direction; outside, for aesthetic and explanatory purposes (apparent motion). Replacing the factory mechanics of the salvaged turntable, a Playstation Eye webcam aiming upwards senses a portion of the gray code disc, the least-weight bits on the outer edge, aided by extra illumination (a backlit USB hub). A computer vision application made with OpenCV and libusb (forking Eugene Zatepyakin’s libusb-based driver \(^55\) reconstructs the bits of slices in realtime so as to recover speed and orientation, or position at low speeds. This code has also been integrated as MediaCycle plugin.

Initially we wanted to make a force-feedback device towards physical effects, but we noticed that the inertia of the platter made it interesting for browsing lengthy media files. An alternative we had in mind was to retrofit a Wiimote gyroscope and accelerometer, making the tangible wireless.

4.2.8.3 Contributions

This setup for tangible intra-media navigation with MediaCycle has been demoed at the ACM conference on Tangible and Embedded/Embodied Interaction \(^56\). Rudi Giot came up with the idea of using Gray code. The author of this thesis repurposed the turntable and developed the accompanying application(plugin).

---

\(^55\) https://github.com/inspirit/PS3EYEDriver

4.2.9 Starfish eNTERFACE: tangible inter-media browsing (2014)

4.2.9.1 Motivation and intention

With AudioCycle, we chose to represent media items in a 2D visualization reminiscent of a galaxy and sorting piles on a desk. A question left open was: how can the expendable, almost infinite space of the digital haystack of media collections, be paired with a user interface that makes digital fishing efficient and pleasurable?

4.2.9.2 Design and prototype

For inter-media navigation, the Starfish eNTERFACE, uses a repurposed 3D force-feedback controller. The now extinct Novint Falcon is mounted in upright position. The device can be either docked on a truss with cell clamps (a piece of metallic structure initially designed for arts stage setups) in a desk/office setting (as pictured in the margin); or gripped to the side of a table with clamps, in a mobile setup (as pictured above during a demo abroad). When positioned under the table, it stays at a certain height and distance so that the user hand can rest on it.
The device is locked on a force-feedback plane as in the previous DeviceCycle experiments but rotated: the knob is blocked on an horizontal plane (otherwise the force sends it upwards), and is either attracted to the corresponding closest media item in the visual space, or is added friction when passing over such items. It thus feels similar to a car gear shift, or a space-shift colliding into space debris. To make it more affordant and comfortable, the knob was replaced by a wireless mouse, as pictured in the margin.

4.2.9.3 Contributions

This setup for tangible inter-media navigation with MediaCycle has been demoed at the ACM conference on Tangible and Embedded/Embodied Interaction 57. Both desk and mobile clamping solutions came up through brainstormings with Willy Yvart. The author of this thesis developed the required software to interconnect the device and MediaCycle, first using DeviceCycle in PureData, then hardcoded in C++ towards a standalone application suitable for user-testing.

4.2.10  AudioMetro: directing search for sound designers (2014)

4.2.10.1  Motivation and intention

Through past prototypes, particularly AudioCycle and AudioGarden, we designed visualization techniques and we intuited these would be efficient and pleasurable. AudioMetro results from observations drawn from user experiments (reported in next chapter) bringing clues on how content-based similarity can complement user interfaces for browsing sound collections.

4.2.10.2  Design and prototype

Our previous prototypes made use of two visual variables to represent content-based similarity: position to induce proximity and color to identify clusters. With AudioMetro we refined our use of position to direct the search pathways in local neighborhoods of sound similarity that are emphasized by glyph representations mapping a perceptual audio feature to color and shape.

4.2.10.2.1  Content-based glyphs as sound icons

For each sound, we mapped the mean over time of an audio feature, perceptual sharpness, to the Value in the Hue Saturation Value (HSV) space of the color of the node representing each sound, normalized against the Values for all sounds in each collection. We also mapped the temporal evolution of perceptual sharpness clockwise to the contour of the nodes. Examples of such glyphs are illustrated in Figure 4.47 featuring clearly similar pairs.
4.2.10.2.2 Local neighborhoods through dimension reduction To position the sounds in a 2D representation, we apply a dimension reduction technique (t-SNE) to an audio feature set (both detailed in 58), to which perceptual sharpness was also added, intuiting it would gather closer items that are already similar visually though their glyph representation, similarity being cued by the same content-based feature.

An undesirable artifact of the t-SNE approach comes from its optimization procedure, which relies on gradient descent with a randomly initialized low-dimensional representation. This creates a stability issue, where several runs of the algorithm may end up in different representations after convergence. This works against the human memory, especially for exploratory search. We solved this issue by choosing to initialize the low-dimensional representation using the two first axes of a Principal Component Analysis (PCA) of the whole feature set.

4.2.10.2.3 A proximity grid optimizing nearest neighbors For the removal of overlap in 2D plots, we borrow a method initially designed to solve the problem of overlap for content-based image browsing: a proximity grid 59. A proximity grid consists in adapting the coordinates of each item of a 2D plot by magnetizing these items on a grid, as in Figure 4.48.

Rodden60 and Basalaj’s 61 doctoral works are heavily cited respectively for usability evaluation in image information retrieval; and for multi-dimensional scaling techniques evaluation; but almost never for their proximity grid approach.


Figure 4.48: Discretizing plot coordinates using a proximity grid.

To our knowledge, no audio browser approached this solution.
To assign items to grid cells, we implemented the simplest spiral search method described in Basalaj’s PhD thesis, named **greedy method**, with the **empty strategy**. We opted for a simplification: the spiral search for empty cells was always turning clockwise and started above the desired cell, while it is recommended to choose the rotation and first next cell from exact distance computation between the actual coordinates of the sound item and the desired cell.

To determine the order of the item-to-cell assignments, we used the fast minimum spanning tree algorithm implementation from the machine learning library **mlpack**\(^{62}\) of Boruvka’s dual-tree based on k-dimensional trees described in \(^{63}\).

To come up with a tradeoff between density and neighborhood preservation, we estimated the number of high-dimensional nearest neighbors (k=1) preserved in 2D at a given grid resolution simply by counting the number of pairs in adjacent cells. Figure 4.50 visualizes the proportion of direct neighbors (compared to the collection size) preserved for different grid side values, comparing a proximity grid (stacked bars) versus a grid ordered by filename (horizontal lines). The minimal side of a square grid is the ceil of the square root of the collection size (leftmost part of the horizontal axis), providing the most space efficient density. To approximate a least distorted grid, the collection size can be taken as grid side (rightmost part of the horizontal axis).

We distinguish the amounts of horizontal and vertical and diagonal neighbors since different search patterns may be opted by users: mostly horizontal or vertical for people accustomed respectively to western and non-western reading order, diagonal may provide a lighter constraint allowing greater grid sides.

The obtained layout can be explained through the metaphor of a metro map: metro stations (sound items) can be close one-another but not necessarily connected by the same metro lines (be part of similar neighborhoods),

---


4.2.10.3 Contributions

This work was presented at two conferences: the 15th International Society for Music Information Retrieval (ISMIR) conference\textsuperscript{64} and the 9th ACM Audio Mostly conference\textsuperscript{65}. The author of this thesis developed the glyph computation and proximity grid optimization; and undertook the user evaluations.


4.3 Organizing our browsers

In this section we isolate or combine traits of the taxonomy described in Chapter 2 to provide different layer of analysis of the state-of-the-art of media browsers, again mostly focused on content-based ones. Table 4.11 compares browsers by media type (\musiconly\, \videoonly\), granularity (\granularity\), organization (\organization\), visualization dimensions (\visualization\) and interaction (\interaction\).

<table>
<thead>
<tr>
<th>Browser</th>
<th>Year</th>
<th>\musiconly\</th>
<th>\videoonly\</th>
<th>\granularity\</th>
<th>\organization\</th>
<th>\visualization\</th>
<th>\interaction\</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioCycle</td>
<td>2009</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>AudioGarden</td>
<td>2010</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>LoopJam</td>
<td>2011</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>MashtaCycle</td>
<td>2013</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>The Listening Room</td>
<td>2013</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>VideoCycle</td>
<td>2013</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>LavaAMP</td>
<td>2013</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>Grayfish Squidget</td>
<td>2014</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>Starfish eNTERFACE</td>
<td>2014</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>AudioMetro</td>
<td>2014</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
</tbody>
</table>

Table 4.11: System-wise browsers comparison

Table 4.11 compares browsers by media type (\musiconly\, \videoonly\), granularity (\granularity\), organization (\organization\), users (\users\), communities (\communities\) and usability evaluation (\usability\).

<table>
<thead>
<tr>
<th>Browser</th>
<th>Year</th>
<th>\musiconly\</th>
<th>\videoonly\</th>
<th>\granularity\</th>
<th>\organization\</th>
<th>\visualization\</th>
<th>\interaction\</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioCycle</td>
<td>2009</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>AudioGarden</td>
<td>2010</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>LoopJam</td>
<td>2011</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>MashtaCycle</td>
<td>2013</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>The Listening Room</td>
<td>2013</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>VideoCycle</td>
<td>2013</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>LavaAMP</td>
<td>2013</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>Grayfish Squidget</td>
<td>2014</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>Starfish eNTERFACE</td>
<td>2014</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
<tr>
<td>AudioMetro</td>
<td>2014</td>
<td>\musiconly\</td>
<td>\videoonly\</td>
<td>\granularity\</td>
<td>\organization\</td>
<td>\visualization\</td>
<td>\interaction\</td>
</tr>
</tbody>
</table>

Table 4.12: User-centered browsers comparison

Our prototyping efforts were more focused on non-musical sounds than video content. We progressively pushed towards tangible user interfaces, always with 2D displays. We aimed at targeting mixed user populations: anybody interested in media installations, media artists, sound designers.
4.4 Conclusion

Our contributions to the MediaCycle framework can be summarized as follows:

• improving the user experience: opening its control to user-definable means, by the implementation of an OpenSoundControl (OSC) and Tangible User Interface OSC (TUIO) bridge, and a small toolbox for the PureData modular framework supporting several off-the-shelf devices;

• not reinventing the wheel: integrating several existing third party libraries, for audio feature extraction (YAAFE), dimension reduction (Divvy, Dredviz), audio rendering (STK and RtAudio), video thumbnailing (ffmpeg and OpenCV), generic thumbnailing (SVG), user interaction (hidapi);

• modularizing its architecture: with more plugin types, granularizing the content-based workflow (from indexing to visualization), some dependent of media types, other generic;

• targeting cross-platform support: pushing MediaCycle beyond Apple hardware and operating systems by switching the GUI from Cocoa to Qt;

• easing the development process: by the implementation of CMake-based building scripts, allowing developers to chose their own IDEs, to make packages or releases for the major operating systems (Debian packages for Ubuntu, .app bundles for Apple OSX, self-contained .exe executables for Microsoft Windows), and to easily start up the development of new applications and plugins by scripted dependency management.

We intuitively understood through such browser prototypes that the benefits from multimedia information retrieval vary along the type of users targeted and the type of media analyzed. For instance MIR might not be felt robust enough for artists wanting to present media collections. Tangible proofs are required to (in)validate such claims: it is the objective of next chapter reporting qualitative and quantitative evaluations.
5 Experiments, evaluations

In the previous chapter, we described many prototypes developed with the MediaCycle framework, with design cues in mind from human-computer interaction, information visualization, content-based organization. Building upon Varèse’s quote, how dare one state such contributions may be novel? Through qualitative and quantitative evaluations. When this thesis was presented in a doctoral consortium, we wanted to explore two research assumptions by evaluating some of prototypes presented in the previous chapter. This chapter targets the first.

- Daily, focused, accurate and repetitive tasks, such as multimedia data mining including sound effect retrieval or video de-rushing, work best with tangible interfaces that allow minimal movements in the gestural space to cover all the media content space.

- Exploratory and collaborative tasks in public settings, such as playing with art installations, can be advantageously performed with free-form interfaces (gestural, contactless).

In this chapter, we first report a contextual inquiry (5.1.1) with sound designers composing collections for LoopJam (4.2.3) and thinking-aloud (5.1.2) over AudioCycle (4.2.1). Introduced to known-item search (5.2.1) through the Video Browser Showdown (5.2.2) with VideoCycle (4.2.6), we applied this evaluation method to sound browsing in 4 iterative experiments (5.2.3) that lead to AudioMetro (4.2.10).

\[\text{EDGARD VARÈSE}^{1}\]

\[\text{Odile Vivier.} \quad \text{Varèse. ISBN: 2-02-000254-X. Seuil / Solfèges, 1973}\]

5.1 Qualitative: contextual inquiries

Eoin Brazil’s master thesis provides recommendations for the design of sound browsers obtained through an interview with sound designers (pages 91-92) \(^3\). We complemented these findings by our observations from a contextual enquiry undertaken with sound designers composing sounds for LoopJam (5.1.1) and thinking-aloud over AudioCycle (5.1.2).

5.1.1 Composing collections for LoopJam

In mid 2012, a contextual inquiry had been undertaken with 4 electroacoustic music composers or sound designers interested in creating collections of loops for the LoopJam installation that would be showcased during 3 weeks for the Citysonic 2012 festival in Brussels, Belgium. Although a call for participation was posted online on the numediart website and newsletter, all of them were recruited through personal contact, as they showed motivation to test MediaCycle at earlier casual meetings. All of them were male, Belgian citizens, of age ranging from 25 to 54. For each of them, an interview was conducted individually within sessions of approximately 1 hour, with several goals in the agenda. First, getting to know more about them and their musical/sounding practices by having them fill in an introductory questionnaire. Second, tutoring them hands-on on how to create collections with a barebones version of the AudioCycle application and experiencing these in the LoopJam installation while gathering their think-aloud comments. Third and last, finishing by a satisfaction questionnaire.

5.1.1.1 Questionnaire on practices

Interviewing such a number of experts on their practices produced some valuable feedback, even it can’t be accounted for to be statistically significant. Here follows a summary of the most relevant questions and answers from the questionnaire.

When asked with which tools or techniques they organize their audio collections, they all checked the same answer: “through the file tree of a file browser (using folders, subfolders, labels)”, among other choices that were “by filling-in annotations in a separate spreadsheet or text file”, “using markers, post-its, bookmarks, tags”, “with (a) dedicated automated organization tool(s)”, plus the possibility to write down a custom answer.

Most of them use Apple computers running the Finder that can be considered as the most advanced and user-friendly file browser, but at the time of the interviews this browser didn’t support user-defined tags (that were introduced later on with OSX Mavericks in 2013), but just colored tags.
 experiments, evaluations  185

Most knew about professional sample libraries organizing tools, such as SoundMiner, but the high license fee prevented them from using it, as freelance workers. We will see in the next qualitative evaluation that these financial constraints change for professionals working in sound design studios.

All of them confirmed to use uncompressed lossless formats to manipulate and compose audio content: WAV and/or AIFF. At that stage and for such users, adding support for lossy formats or compressed lossless formats was thus not necessary. Regarding the storage medium, all of them favored external hard drives for their own creations, older technologies such as magnetic tapes solely used for their sound print as effect.

This questionnaire was later on generalized to more media types (not only audio, but visual media too), complemented with more questions and posted online using the open source LimeSurvey survey engine 4.

5.1.1.2  Feedback from the tutorial

Quoting 5, 5 users should be sufficient to grasp 80% of the most common usability issues of our framework for a given media type through one application. These ranged from simple issues, such as bugs or opting for generic shortcuts widely used in other multimedia applications, to more intricate obstacles in the workflow of the application.

Here we report only general concerns that may be useful for building similar browsers.

5.1.1.2.1  Major strengths

• LU: the modular aspect of the user interface (closable panels)
• JLP: automatic organization (AudioCycle) and ease of use, conviviality for LoopJam
• JLP: “MediaCycle offers a simple means to generate sound snippets”
• RG: graphic representation in clusters
• RG: immediate access to the sounds
• RG: display of the waveform directly in the browser
• GK: simple to use, good feeling, intuitive, instant reactivity

5.1.1.2.2  Main weaknesses

• JLP: missing realtime recording of the instant navigation / manipulation audio rendering
• JLP: missing export of ordered lists to the Finder
• JLP: (minor) changing audio mode presets without restarting the rendering (solved)
• LU+RG: missing support of tags/taxonomies (tag support added later)
• RG: missing support for removing sounds from the collection directly in the browser (later added)
• RG: browser waveforms are not interactive, i.e. the playback can’t be altered by manipulating these

5.1.1.2.3 Interactivity of LoopJam
• LU needs more modes of playback than audio hover (one-shot play), defined per sound
• JLP needs synchronization/sampling modes assigned per sound
• LU and GK would want a solo version
• LU would like to substitute the free-form interaction (gestures sensed by a Kinect RGB/D sensor) by a touch-based interaction on a separate device (fingers on an iPad), while a close solution was presented (fingers on a multitouch trackpad)
• RG at first sight of a Kinect, users may expect a similar user experience to Kinect-based games, more fully-fledged
• RG felt that filenames should be hidden in an interactive installation setting
• RG recommends the support of spatialized sound for a more immersive experience (later implemented)
• RG would have preferred that the banner providing information on the collection and its curator would be deported on an extra screen, similarly to most exhibitions, rather than residing on the bottom of the main screen, what was a reference to the overlays on music videos in music TV channels
• RG would like another synchronization/quantification mode similar to the behavior of a sequencer application for live music, Ableton Live, where triggered sounds actually start at the upcoming measure, and don’t stop before the ending of a measure

5.1.1.2.4 AudioCycle
• LU would appreciate the support of sound effects plugins
• LU would like a legend describing the assignment of colors to clusters. Problem: unsupervised classification, only the number of clusters can be user-defined, so there is no automatic labeling possible at that stage
• LU would want to annotate sounds with keywords describing their texture
• LU would like to manually segment sounds to look for similar segments
• GK would like to manually position start and end playback limits inside sounds
• LU would want a visual normalization of the displayed waveform (actionable on demand: inactive when comparing sounds, active when focusing on one sound)
• LU while timeline waveforms are interactive, browsing waveforms on the inter-media browser would be useful as well (ZUI)
• LU would want more control over the general graphical rendering of the application (contrast, background color)
• LU recording the instant sound output would be useful
• LU exporting the collection as a folder (like sequencers) would be useful = JLP export an ordered list
• LU and RG comparing the use of a given sound between different composition projects or different users
• LU recording the evolution of browsing parameters through a live composition = JLP keep a timeline/log of the actions of the user
• LU selecting a subset of the collection to refresh the view (sub clustering)
• RG “sound designers would not consider a burden to annotate sounds if it helps their organization”
• RG would want features plugins to allow the organization per recoding time and location (similarly to EXIF metadata recorded in photo files by most cameras)
• RG raised a maximum of the flaws affecting widgets (useless menus, menus that could be merged to remove the distinction between files and collections that lead users to activate wrong menus), engineering-driven names (the application name was the same as the folder containing its source code, bearing acronyms of some third parties, what is irrelevant for users), and shortcuts (for instance Escape is expected to leave the fullscreen mode)
• RG suggested that every palette should be listed in the Display menu to show/hide these efficiently without knowing the standard tricks (later implemented)
• RG would want to remove sounds my mass filtering these from a spreadsheet with name, path, etc...
• RG suggests that a palette servers as realtime help/tips depending on the actions of the user (similarly as in Ableton Live)
• GK would like to control sound modification cues per sound: volume, pitch, reverse, constant looping
• JLP would need the ability to export sounds to ProTools
• GK would like several pages/thumbnails of sounds, macro collections that are switchable without loading time
5.1.1.2.5 Other applications

- LU complements the Finder by replacing its simple audio preview featuring only a slider and playback controls (activated by pressing the space bar when a sound is selected) by Audio Snapper by Audio Ease, a third-party application that adds a visualization of the waveform to the preview and manual segmentation for drag and drop.

- RG tried EAnalysis by Pierre Couprie and Acousmographe by INA/GRM, offering a reverse approach compared to MediaCycle: these allow to annotate musical pieces with different taxonomies, for instance Pierre Schaeffer’s sounding objects, inspiring perspective for MediaCycle.

5.1.1.2.6 Organization

- LU: clustering sounds in short vs long duration groups would be handy. As immediate answer, the Gramophone view of AudioGarden, a polar display with inverse duration on radius and a content-based feature on angle, felt like a nice first step.

5.1.1.2.7 Concerns

- LU: a convincing fast playback for long sounds is missing.
- JLP: realtime recommendation of similar sounds is missing.
5.1.2 Media workflows of sound designers

From late 2012 to mid 2013, we undertook 3 think-aloud sessions with a sound designer from a Belgian studio working on national and international movies, partnering in a project to adapt MediaCycle for sound design settings. The meetings consisted in having the expert show us his working setting and techniques, and experience iterations of our prototypes.

5.1.2.1 Description of a typical sound design setting

Figure 5.1 illustrates a typical setup for sound design.

The practitioner is explaining how he produces sound design for movies on his daily workspace. A multi-screen setup allows him to assign to each different modes of navigation: the bottom screen shows the timeline of a sequencer where sounds are positioned horizontally according to the storyline and vertically in tracks depending of their category; the top screen maximizes the media manager resembling a spreadsheet application that allows drag and drop from/to the timeline. The desk is surrounded by loudspeakers to reproduce a standard spatialization setup. Besides the usual keyboard and mice, a fader box allows to rapidly modify effects parameters associated to the different tracks. Trackballs offer an alternative to the mouse for panning and selection tasks, so as to compensate the muscular fatigue effected differently by each device (mostly the wrist is affected when using the mouse, and the thumb with the trackball).
5.1.2.2 Clues in sound manipulation influencing sound organization

We report the discussions emerging from the interview that affect sound organization.

5.1.2.2.1 Sound production

There are many ways to produce desired sounds: by recording them with actual objects or cueing artifacts (for instance the use of coconuts to mimic horse steps), by synthesis (where for instance physical modeling is a sought technique when naturalness is desired) or from existing sound recordings available in sound libraries. We target the last case.

5.1.2.2.2 Sound compositing

A technique to create sounds corresponding to physical actions of elements from the scene consists in “piling” several sound snippets, offering a high degree of freedom to tailor the sound to the desired visual identity. For instance for the case of a broken tree falling, several temporal and mechanical steps bear each a sound identity: the trunk breaks, the tree falls, leaves shatter, the tree contacts with the ground.

5.1.2.2.3 Sound identity

In the case of series or movie franchises with sequels, sound sample subsets are often reused to maintain an aural identity. This calls for the analysis of the usage of each sound beyond the scale of a session or project.

5.1.2.2.4 Sound categories (and colors)

Timeline tracks and mixing tracks in sequencers such as ProTools are added for each group of sounds requiring specific effects (background sounds may be grouped, specific action-related sounds may stand alone on one track) and can be tagged with colors. Such colors can be assigned to families of sound textures (for instance: ambience and background noise, effects, and so on...). The initial idea of assigning color to clusters in MediaCycle complies to such a usage. Flashy colors such as the trademark fluorescent yellow are also used to draw attention, for instance when reviewing projects in progress.

5.1.2.2.5 Sound context

For any of the aforementioned sound manipulations, sound samples first need to be found according to the context. For this purpose, sound designers usually first enter a textual query describing the desired sounds in a file browser or a dedicated sound organizing tool, to narrow down the search from the whole collection of sounds. At that stage a list is presented, often as a spreadsheet with associated metadata on columns. Then the sound designer can reduce the size of list by other iterations of filtering by tags, up to being left to listen to the results in a sequential order as illustrated in Figure 5.2.
Most sound libraries, bought from editing companies or built by sound designers themselves, are nowadays always tagged, however with various levels of definition. We can consider that at least the first depth of textual filtering can not be outperformed by other methods. But after this very last phase, when the search can not be refined with more terms, sound designers could take advantage from aided organization beyond words: using a content-based approach.

We compiled altogether an idealistic list of features that would allow a flexible organization of sounds for sound design:

- distance between the sound source and the microphone
- location of recording (with subset precisions such as indoor vs outdoor)
- nature of the source, including its material
- type of action (for instance “closing” or “squeaking” door)
- perceptual/emotional criteria (dark vs bright, aggressive vs calm)
- duration (impact sounds are short, ambience sounds are longer)
- ascending/descending/stable pitch
- homogeneous vs dynamic
- tonality and harmonicity vs noisiness
- the variations in the envelope

**Figure 5.2:** Search results in a sound browser for sound design – © Matthieu Michaux
5.1.2.3 Think-aloud test of AudioCycle with various layouts and features

One of the requirements elicited from the previous qualitative evaluation of AudioCycle with electroacoustic music composers reported in Section 5.1.1 was the interaction between MediaCycle applications and external applications for media manipulation, such as audio sequencers. For this purpose generic drag and drop support was implemented.

This time, we had the sound designer test a newer AudioCycle application packing such new features and several positioning algorithms: the Student-t Stochastic Neighbors Embedding (t-SNE) presented in \textsuperscript{10}, the first two axes of a Principal Components Analysis (PCA), and the Gramophone polar view (radius: inverse duration, angle: content-based feature) introduced for project AudioGarden. We reproduced the workflow that we have just described in the previous section: using a file explorer (Apple Finder on OSX), we filtered the One Laptop Per Child (OLPC) Sound Library \textsuperscript{11} (see section 5.2.3.2) with two of the most occurrent tags (“water” and “metal”) and dragged and dropped the results into AudioCycle.

Here follow the main concerns that we obtained from the think-aloud test: perceived organization, representative and human-readable features, content-based filtering tools, then other tracks of investigation.

5.1.2.3.1 Perceived organization

When presented with a view the first two axes of a PCA, the sound designer felt that the organization was random.

5.1.2.3.2 Representative features

While testing the Gramophone polar view, one observation that emerged by a discussion between the parties is that the duration as extracted from file metadata isn’t always accurate as feature for the display: sound files may contain silence between sound events. This leads to several content-based solutions, such as segmentation and silence detection.

5.1.2.3.3 Human-readable features

This is the first user test during which weights of features are modified. In the past these were prepared beforehand and fixed during the test sessions. The observers tried to tailor the organization to the sound designer’s recommendations, one being to better separate short and long sounds, for instance by keeping only spread (or variance) as statistic summarizing the chosen temporal features (removing skewness and kurtosis).


\textsuperscript{11} http://wiki.laptop.org/go/Sound_samples
There is a need for features that are self-understandable from non-experts in digital signal processing. Some names of audio features may appear cryptic to sound designers, for instance “Delta MFCC: Centroid” or “Spectral Flatness: Kurtosis”, what may prevent these users from choosing a relevant selection of features to refine their search. To solve this issue, the tracks of Grill et al. can be followed, by replacing such audio features by personal constructs \(^{12}\).


5.1.2.3.4 Content-based filtering tools

The observers thought of proposing tools inspired by vectorial image editing and sound editing applications. For instance a “content-based rubber” that would delete the k-nearest neighbors of the hovered sound, or a “content-based crop” that would filter the view by selecting sounds as boundaries.

5.1.2.4 Other tracks of investigation

Several needs that go beyond the scope of this thesis were elicited.

5.1.2.4.1 Sound separation

Source separation algorithms would be welcome to split field recordings into the various aural contributions of the background ambiance and located events. A corollary, rather than fragmenting the sound into sources, would be to cue in the sound, untouched, at the time when events occur, to synchronize it with the action of the movie.

5.1.2.4.2 Sound annotation

Imagining an online service for sharing and distributing sounds, users annotating sounds with tags so as to improve their organization could be encouraged to do so by earning virtual points, increasing their “reputation” as in social networks, for instance Social Sound Design \(^{13}\).

\(^{13}\) http://www.socialsounddesign.com
5.2 Quantitative: known-item search tasks

After qualitative evaluations of our work based solely on interviews and debriefs, we wanted to investigate complementary ways to evaluate our media browsers, by logging quantitative metrics from simulated tasks. We learnt about Known-Item Search tasks (5.2.1) at the Video Browser Showdown with VideoCycle (5.2.2). We applied the method to AudioCycle (5.2.3).

5.2.1 Known Item Search (KIS) tasks

Known-item search tasks consist in displaying a media element or fragment to a tester and requiring her/him to find back the target inside a collection or long record. Task success and retrieval time can be measured and serve as metrics.

The Great CHI’97 Browse-Off forum \textsuperscript{14} can be considered as a pioneering evaluation of browsers in a live competition setting. 16 contestants (half novices, half experts) tested 4 different browsers. The Hyperbolic Tree Browser outperformed the others, followed by Microsoft’s File Explorer. The authors of the first reproduced a larger evaluation to validate this result, this time comparing only these two browsers, some of the experiments invalidating the contest results \textsuperscript{15}. Its differs from our scope since their dataset contained hierarchical structured data.

5.2.1.1 For video browsers

The TRECVID evaluation of video browsers appeared in 2002 \textsuperscript{16}, building upon text retrieval research, and has been including known-item search tasks since 2010, but with text-only descriptions of targets, and not taking user interaction into account.

VideOlympics \textsuperscript{17} \textsuperscript{18} took over from 2007 to 2009, still through text-only descriptions, but promoting advanced visualization and interactivity employed in the challenging systems.

Since 2012, Schoeffmann and Bailer have been organizing a yearly interactive evaluation of video browsers through known-item search tasks in a competitive ambience, as special session of the Multimedia Modeling conference, later promoted to collocated workshop: the Video Browser Showdown \textsuperscript{19}.


\textsuperscript{17} Cees GM Snoek, Marcel Worringer, Ork de Rooij, Koen EA van de Sande, Rong Yan, and Alexander G. Hauptmann. “VideOlympics: Real-time evaluation of multimedia retrieval systems”. In: IEEE Multimedia 15.1 (2008), pp. 86–91

\textsuperscript{18} http://www.videolympics.org

\textsuperscript{19} http://www.videobrowsershowdown.org
Surprisingly, from an evaluation by the organizers of the results of the 2012 edition 20 that preceded our participation, it appeared that a simple browser which is not content-based produced the best results. This motivated us to participate to its 2013 edition, since at the time the MediaCycle framework only included state-of-the-art techniques from multimedia information retrieval applied to video, and since the author of the current thesis wanted to focus on interactivity and visualization.

Some video browsers presented in the second chapter of this thesis were evaluated through known-item search tasks. In a subsequent publication, Haesen et al. explained that they did not record the time of execution of search tasks since criteria for searching video fragments are generally related to personal experience and vary along people 21.

5.2.1.2 For audio browsers

When reviewing results of queries past keyword filtering, sound designers may not picture target sounds accurately in their head or be able to name further characteristics, leaning closer towards exploratory search. However a method to quantitatively estimate the efficiency of a sound browsing system is required to assess if research efforts are in the right tracks.

MIREX 22, the equivalent evaluation of TRECVID for music, hasn’t yet been proposing KIS tasks so far aside other tracks.

Font’s work about sound browsers compared layouts for sound browsing: automatic (PCA), direct mapping (scatter plot) and random map. It deliberately rejected investigating on time and speeds, claiming people have different search behaviors 23, what mirrors the aforementioned argument from Haesen et al.

Recently, key researchers from the music information retrieval community have been criticizing the sole use of p-values as satisfactory goal to reach statistical significance in quantitative tests, while effect size and the magnitude of the difference between compared results should also be considered 24.

A few audio browsers presented in the second chapter of this thesis were evaluated through known-item search tasks.


5.2.2 Competitive tests at the Video Browser Showdown

We participated to the second edition of the Video Browser Showdown. For that edition, one track was proposed: finding segments of videos, the target displayed once on a screen, with expert and novice runs. Figure 5.3 captures the ambience of the contest and its setting.

![Setting and ambience of the Video Browser Showdown (VBS) at the Multimedia Modelling conference (MMM) in 2013 – © a MMM 2013 volunteer](image)

5.2.2.1 Setup

Figure 5.4 shows a close-up of the screen, that not only displays target segments, but also realtime scores of each browser (one per column), with cells in red when submissions (through the HTTP protocol) are false, green when successful.

![Picture of the Video Browser Showdown (VBS) server displayed on a large TV screen – © Klaus Schoeffmann](image)
5.2.2.2 System

For that contest, we proposed a tangible user interface \(^25\). The version of VideoCycle that took part in the challenge is a subset of the application initially submitted, focusing on visualization of and interaction in a video timeline.

5.2.2.2.1 Visualization Display had been restricted to a video timeline featuring from bottom to top in Figure 5.5:

- (bottom) a summary row of keyframes evenly-distributed over the whole video, maximized to the screen width;
- (in between) a selection row with keyframes corresponding to the duration range determined by the width of the selection (green markers) centered around the playback cursor over the aforementioned summary row;
- (top) a large playback view of the video.

5.2.2.2 Interaction In addition to the usual control offered by a mouse or trackpad, the span of the selection could be modified by the inner wheel of a jog wheel and skipping frames by its outer wheel, while the segment boundaries could be submitted to the VBS server by pressing a button of the jog wheel (left insert in Figure 5.5). Keyframes from the selection row could be clicked to reposition the playback cursor to the corresponding frame.


Figure 5.5: Screenshot of the VideoCycle version that contested in the Video Browser Showdown – © Christian Frisson.
By assigning the jog wheel to the left hand and the mouse to the right hand (or their permutation depending on the preferred hand), we posited that the user may keep a better motor memory of the controllers (as opposed to keyboard shortcuts), thus reducing head movements to look frantically at the controllers during the search, and focus the attention on the screen.

5.2.2.3 Suggested search behavior

No user command logging was recorded during the tasks. But the participant used a repeatable browsing technique for each task, comparable to exhaustive search: linearly skimming through the video using the cursor while looking at the selection keyframes row, increasing its span when a matching scene was discovered, if not skimming the video again.

In the previous VBS 2012, a submitted segment was considered as correct if \((S - 125) < s < e < (E + 125)\), where \(s\) and \(e\) are the submitted start and end frame numbers and \(S\) and \(E\) are the begin and end frame numbers of the target clip. Therefore, the participant did not try to set the time boundaries of the segment carefully before submitting it to the server, but rather made sure that the current playback time would be in between, as this value would be submitted for both boundaries.

5.2.2.4 Design

During a 2-hour session of the Multimedia Modelling 2013 conference, 6 browsers were evaluated in two runs: one expert run with browsers manipulated by their creators over 10 tasks, one novice run with volunteers from the audience randomly assigned to each browser over 6 tasks. Each task consisted in finding a 20-second segment displayed once at the beginning of each task, randomly chosen from a given video among a corpus of news broadcasts each more than 1 hour-long that were transferred to participants approximately 1 month before the session. Videos were given access by the EBU MIM/SCAIE project of the Flemish public broadcasting organization VRT, Belgium. Textual analysis and queries were forbidden to focus the comparison on interaction rather than data mining.
5.2.2.5 Results

Figures 5.6, 5.7 and 5.8 are borrowed from the evaluation paper co-authored by all contestants physically present at VBS 2013 28. We refer to this paper for a comprehensive evaluation. Here we summarize our observations focused on VideoCycle.

During the expert run, the author of this thesis noticed that VideoCycle submitted intervals of time instead of frame to the server and corrected the mistake. This explains the low scores of VideoCycle visible in Figure 5.6. The organizers allowed the author to remain tester of VideoCycle in the novice run.

In Figure 5.7, from the right graph for the novice run, we can notice that a similar average amount of correct submissions was obtained with all browsers. Wrong submissions were quite numerous for VideoCycle: content-based abilities may have helped to discriminate similar segments.

Figure 5.6: Total scores per browser at the Video Browser Showdown 2013 – © Schoeffmann el at.

Figure 5.7: Successful/failed submissions per browser and novice/expert run at VBS 2013 – © Schoeffmann el at.

From Figure 5.8 we can notice that success times (the duration between the beginning and the time in submitting the correct target) are quite spread apart the median. This may be due to the inefficiency of the brute-force search technique used for the tasks.

![Success times per browser and novice/expert run at VBS 2013](image)

**Discussion**

The system we employed for the competition didn’t make use of content-based similarity. This browser offered more advanced visualization and interaction techniques than a simple video player with a seeker bar, common in many desktop and web-based applications. Such a kind of player has been used as baseline system for additional tests described in the evaluation paper. Our system can be considered as a baseline for rapid serial visual presentation. We believe that such an evaluation should feature various baseline systems to evaluate the distinct contributions in performance between interface design (HCI) and content-based analysis (MIR).

After this evaluation, Schoeffmann and Cobârzan undertook user experiments to assess the usability of such simple video players with seeker bars. They showed that progressing forwards along the video time by dragging the seeker bar with the playback paused was the most efficient in terms of time spent in the search task, but not when dragging backwards as done with VideoCycle at VBS 2013.

---

5.2.3 Controlled experiments for audio content

From our observations during the qualitative contextual inquiries, we understood that sound designers source sounds in massive collections, heavily tagged by themselves and sound librarians. If a set of sounds to compose the desired sound effect is not available, a Foley artist, that can be the sound designer herself/himself, records the missing sound(s), and both will tag these recordings as accurately as possible, identifying many facts from physical (object, source, action, material, location) and digital (effects, processing) properties. When it comes to looking for sounds in such collections, for each query, once successive keywords helped the user to filter down the results, but attained a limit, hundreds of sounds are left to be reviewed.

We elicited the following research questions:

• Can content-based organization help once filtering sounds by tag becomes limitative?
• Are there different search behaviors among users in 2D presentations of results?

We show through 4 within-subject summative user evaluations by known-item search in collections of textural sounds that a default grid layout ordered by filename unexpectedly outperforms content-based similarity layouts resulting from a recent dimension reduction technique (Student-t Stochastic Neighbor Embedding), even when complemented with content-based glyphs that emphasize local neighborhoods and cue perceptual features. We propose a solution borrowed from image browsing: a proximity grid, whose grid side we optimize for nearest neighborhood preservation among the closest cells. Not only does it remove overlap but we show through a subsequent user evaluation that it also helps to direct the search. We based our experiments on an open dataset (the OLPC sound library) for replicability.

5.2.3.1 Choice of software solutions for statistical analysis

Quantitative data was pre-processed and visualized using Octave\textsuperscript{30} in version 3.8.1 with its statistics and plot toolboxes and \LaTeX tikz\ plot export. Statistical evaluations and qualitative data visualizations were performed with R\textsuperscript{31} in version 3.0.3 inside the RStudio development environment\textsuperscript{32}. Both tools combined stand for open source equivalents to the widespread commercial application Matlab by Mathworks, to make the analysis reproducible at no cost. All log datasets and source code files are made available as digital appendices to this manuscript, so that these can be double-checked at will, and may become helpful in the comparison with other browsers.

\textsuperscript{30} http://www.octave.org
\textsuperscript{31} http://www.r-project.org
\textsuperscript{32} http://www.rstudio.com
5.2.3.2 An open dataset for audio collections

The One Laptop Per Child (OLPC) sound library\(^{33}\) was chosen so as to make the following tests easily reproducible, for validation and comparison perspectives, and because it is not a dataset artificially generated to fit with expected results. It is licensed under a Creative Commons BY license (requiring attribution), while current datasets from MIREX face copyright issues\(^{34}\). The library was first aimed at being offered as audio content packed with low-cost computers to be loaded in computer music applications, for educational and recreational use in developing countries were digital literacy is scarce, pushed by a non-profit project started by several US universities and companies. In over 8GB of digital storage space, it contains 8458 sound samples at a sampling rate of 44,1 kHz and a resolution of 16 bit, mono or stereo files. Sound designers collection nowadays contain 10 or 100 times more. Several volunteers, from students in music theory/recording at The Berklee College of Music to linux audio opensource applications creators and users, contributed 90 sub-libraries, combining diverse types of content or specializing into one type, among which: riffs or single notes from musical instruments, field recordings, Foley, synthesized sounds, vocals, animal sounds.

Figure 5.9 proposes a tag cloud of the keywords present in the filenames of all sounds, split into words at each single capital letter and by rejecting non alphabetical characters, without spell check, produced using Jason Davies’ Wordle-inspired word cloud layout generator\(^{35}\). Even though tables outperform tag clouds in efficiency\(^{36}\), this summarization provides a biased but cost-effective overview of the collection, without diving into complex semantic analysis, and emphasizes some keywords relevant to be used as queries when replicating a repeated task of the workflow of a sound designer: first filter by tag(s), then browse to review and refine.

---

\(^{33}\) [http://wiki.laptop.org/go/Free_sound_samples](http://wiki.laptop.org/go/Free_sound_samples)


\(^{35}\) [http://www.jasondavies.com/wordcloud/](http://www.jasondavies.com/wordcloud/)

These words can be sorted into categories as illustrated with some salient examples in table 5.1.

<table>
<thead>
<tr>
<th>Category</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>instruments</td>
<td>guitar, p[ia]no, snare, synth, v[io]la</td>
</tr>
<tr>
<td>instr. techniques</td>
<td>[col] legno, gliss[ando], leg[ato], tremolo</td>
</tr>
<tr>
<td>sound objects</td>
<td>bell, box, car, leaves, spring, tin, toy</td>
</tr>
<tr>
<td>sound sources</td>
<td>noise, sfx, vocal</td>
</tr>
<tr>
<td>sound actions</td>
<td>bounce, hit, pop, rub, scrape, snap, tap</td>
</tr>
<tr>
<td>sound effects</td>
<td>dist[ortion], fuzz, rev[erb]</td>
</tr>
<tr>
<td>sound attributes</td>
<td>bpm, long, low, short</td>
</tr>
<tr>
<td>material/matter</td>
<td>air, glass, metal, water</td>
</tr>
</tbody>
</table>

It is to be noted, especially for subset libraries curated by Berklee containing Foley sound design material, that within a given subset most samples seem to have been recorded, if not named, by a same author per subset. It is thus frequent to find similar sounds named iteratively. These are likely to be different takes of a recording session on one same setting of sounding object and technique. Ordering search results by tag filtering in a list by path and filename as would a standard file browser do, will thus imprint local neighborhoods to the list.
5.2.3.3 Experiment 1: content-based cloud

We first wanted to investigate how content-based organization would assist browsing sounds. We compared two layouts: cloud versus grid. With cloud, similar objects are close to one another. With grid, objects are positioned in lexicographic order of their file name, from left-to-right, and then from top-to-bottom.

5.2.3.3.1 System

A usual multimedia information retrieval workflow consists in several steps, that can be performed offline such as indexation and clustering, and almost realtime such as dimension reduction and changing parameters of a visual representation.

A first step is feature extraction. We based our selection of features from the work of Dupont et al. since their evaluation considered textural sounds. In short we used a combination of derivatives of and statistics (standard deviation, skewness and/or kurtosis) over Mel-Frequency Cepstral Coefficients (MFCC), a regular feature in most MIR systems, and Spectral Flatness (SF) correlated to noisiness. Grill et al aimed at defining features correlated to perceived characteristics of sounds that can be named or verbalized through personal constructs, for instance high-low. One application of their work is to simplify the user interface of MIR systems by making the choice of features more understandable by users not expert in signal processing. Following their results, we also made use of Perceptual Sharpness that is highly correlated to the perceived brightness of the sound and that was the most correlated to one feature present in the YAAFE feature extraction library. Since our test collections feature textures of short length and steady homogeneity, we decided not to segment the sounds.

Another step is dimension reduction. We opted for Student-t Stochastic Neighborhood Embedding (t-SNE). In short, this method aims at preserving high-dimensional neighbors in a lower-dimensional projection (here 2D) by estimating the probability of each pair of sounds to be neighbors. Through informal tests we noticed an emergent property of this technique applied to textural sounds: takes recorded from the same source with slight variations are almost always neighbors in 2D.
As input we used a linear combination of all the aforementioned features with equal weights. An undesirable artifact from t-SNE is that the 2D positions are initialized randomly, making the representation of a given sound collection variable over time, what works against human memory especially for exploratory search. We solved this issue by choosing to initialize these positions with the two first components of a Principal Component Analysis (PCA) of the features, probably not providing an optimal solution, but making it repeatable.

Displaying such a representation results in a scatter plot or starfield display. Neither waveforms nor filenames were displayed, to have testers concentrate the visual memory on layouts, and to avoid taking into account the time spent in understanding tags semantically.

In their paper about collection understanding that they oppose to information retrieval, Chang et al. argue how scrolling can be a time-consuming and burdensome interaction technique. We therefore chose to disable panning and zooming.

5.2.3.3.2 Apparatus

Tests were performed on an 15” Apple MacBook Pro laptop (late 2008 model) with a resolution of 1440×900, and the test application was always set to fullscreen. For auditory display an Echo AudioFire 4 soundcard wired to a pair of Genelec 8020 CPM powered loudspeakers were used. A 3Dconnexion Space Navigator 3D mouse was repurposed as “buzzes” to submit the closest sound to the pointer as target by bumping on the device, instilling a game feel. The number of fingers sensed by the multitouch trackpad was accessed through the Apple OSX Multitouch private framework: a 1-finger touch activates the looped playback of the sound represented by the closest node to the pointer/finger.

5.2.3.3 Participants

Testers were recruited from colleagues, experts in computer vision or speech audio analysis, working in a digital signal processing lab. Most are knowledgeable of dimension reduction techniques (some work with PCA) and scientific visualization.


Otherwise noted, this apparatus is maintained in subsequent experiments.
Among the 19 testers (2 females, 17 males), 2 were known to be color blind.

5.2.3.3.4 Design

This controlled laboratory experiment was set up in a small meeting room, free of visual clutter.

Each session was identical for all testers and included sequentially: 1) warming up with 4 simple known-item search tasks to get acquainted with the setup; 2) performing a larger set of 6 known-item search tasks, with times and mouse path logged, while observers collected first impressions aloud from the testers.

Each task was time-limited to 60 seconds, passing automatically to the next after the deadline. The pointer was programmatically moved to the left mid-height position, systematically at the beginning of each task, near a countdown showing the time left to perform the task. Each task was associated to one in 6 collections of 64 elements, filtered by tag from the whole OLPC library, using the following keywords: ball, bell, glass, metal, scrape, water, as illustrated in Figure 5.10. A square integer was chosen as collection size to make sure that the grid visualization would be balanced, symmetric, regular and neutral; quoting visual techniques from Vanderdonckt et al. \textsuperscript{42}. We assumed that a layout without empty grid intersections would be less sensitive to induce pathways, for instance another collection size making the last row unequal in regard to the others might stimulate users to browse the view in the western reading order, from the top-left corner. To trim down the search results that would exceed 64 items for some of the keywords, files presenting iterative patterns in their filenames were first dismissed, since these would be similar “recording takes” as stated earlier. This reduces similarity neighbors artifacts from filename ordering. Collections were displayed either in a grid or a cloud, in a permuted order. The targets were chosen offline without qualitative evaluation by their index import number in regard to the collection so as to be spread over the collections: these targets weren’t heard or visualized during the preparation, so as not to favor any task.

Figure 5.10: Visual organization and target location for each test 1 task, pairing subsets filtered by keyword and layouts (left to right, top to bottom): ball grid, bell cloud, glass grid, metal cloud, scrape grid, water cloud. The target is circled in green.

Figure 5.11 shows the exact screenshots of the fullscreen application for tasks 1 ball grid and 2 bell cloud. The user interface was stripped down to its bare minimum. A column was added to the left, displaying the countdown reset at each task automatically, its background color turning green if the submission was successful, red otherwise, here black.

Table 5.3 provides the filename and OLPC collection for each target, for further investigations.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Target</th>
<th>Texture</th>
<th>Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>ball</td>
<td>Berklee44v12/ball_tunnel_1.wav</td>
<td>impact</td>
<td>2</td>
</tr>
<tr>
<td>bell</td>
<td>Berklee44v11/bell_roll_2.wav</td>
<td>sustained impact</td>
<td>13</td>
</tr>
<tr>
<td>glass</td>
<td>Berklee44v11/glass_tone2.wav</td>
<td>friction</td>
<td>5</td>
</tr>
<tr>
<td>metal</td>
<td>Berklee44v8/metal_swirl1.wav</td>
<td>sustained impact</td>
<td>14</td>
</tr>
<tr>
<td>scrape</td>
<td>PrentRodgers44/SpringDry_scrape10.wav</td>
<td>impact</td>
<td>1</td>
</tr>
<tr>
<td>water</td>
<td>Berklee44v10/watersplash1.wav</td>
<td>sustained impact</td>
<td>27</td>
</tr>
</tbody>
</table>

Table 5.3: Filenames of the targets for each task of test 1
5.2.3.5 Results

From the temporal and spatial variables logged, many quantitative metrics could be extracted, here we report an analysis on: the percentage of succeeded/failed tasks, the time until the target was successfully found when “buzzed” (success times), the distance browsed for each task, the average speed of each task, patterns in the mouse paths.

Before producing statistics over the logged data, so as to select the relevant methods, we should first evaluate if the population of users sampled for the test fit to or differ from a normal distribution.

We introduce a user score allowing to compare the results of all users for each view with vectors of metrics of equal sizes, whether or not each target has been successfully retrieved. Such a score is defined as $\sum t_{\text{max}} - t_i$ where $i$ is the number of tasks, $t_{\text{max}}$ is the time limit, $t_i$ the time of target finding at task $i$.

The Shapiro-Wilk test statistic ($W$) and its $p$-value can be computed to evaluate the normality of distribution of results: grid scores look normal ($W=0.98$, $p=0.94$), cloud scores more uniform ($W=0.91$, $p=0.07$). Normality is not assumed for at least one of the two paired groups, plus we have a small sample size ($n<25$), we will use Mann-Whitney’s u-test for further comparisons.

Figure 5.12: Histograms of scores per user with the grid and cloud views for test 1
Table 5.6 reports the results of two-sample u-tests of the null hypothesis of equal medians of these variables between both views. From this table, we can notice that p-values are quite higher than 0.05, showing that results per variable do not drastically differ between views.

<table>
<thead>
<tr>
<th>Variable</th>
<th>p-value</th>
<th>Z</th>
<th>M_{grid}</th>
<th>M_{cloud}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success time (s)</td>
<td>0.66</td>
<td>-0.43</td>
<td>30.72</td>
<td>32.03</td>
</tr>
<tr>
<td>Stumble time (s)</td>
<td>0.67</td>
<td>0.42</td>
<td>19.59</td>
<td>18.95</td>
</tr>
<tr>
<td>Recollection time (s)</td>
<td>&lt;0.05</td>
<td>-1.96</td>
<td>10.51</td>
<td>15.17</td>
</tr>
<tr>
<td>Distance (cells)</td>
<td>0.95</td>
<td>-0.06</td>
<td>9.19</td>
<td>9.03</td>
</tr>
<tr>
<td>Speed (cells per s)</td>
<td>0.37</td>
<td>0.89</td>
<td>0.31</td>
<td>0.32</td>
</tr>
<tr>
<td>Discovers (sounds)</td>
<td>0.17</td>
<td>-1.38</td>
<td>58.16</td>
<td>69.84</td>
</tr>
<tr>
<td>Hovers (sounds)</td>
<td>0.11</td>
<td>-1.58</td>
<td>33.38</td>
<td>38.68</td>
</tr>
</tbody>
</table>

More tasks were successful with the grid (45/57) than with the cloud (37/57): clues might be present in the pathways browsed by the mouse pointer and the localization of each target in each task. Some tasks were missed for both views: this test can be considered difficult. Successful targets were submitted on average after 30 seconds, which is almost the average time (32s) taken for listening the first second of each sound (64) once and sequentially (64s). There doesn’t seem to be an effect of the choice of layout on the average speed of browsing.
Figures follow to compare several variables, logged or extracted, between views: times required to locate the target (Figure 5.14), times when targets were first stumbled upon (Figure 5.15), recollection times being the difference between the two latter (Figure 5.16), distances browsed (Figure 5.17), average speeds of browsing (Figure 5.18), numbers of sounds hovered cumulatively (Figure 5.20) or discovered at least once (Figure 5.19).

Even if this pilot test wasn’t designed with enough tasks and/or users to be statistically-significant, some conclusions can still be drawn, starting from successful tasks and times:

- More tasks were succeeded with the grid (45/57) than with the cloud (37/57): clues might be present in the pathways browsed by the mouse pointer and the localization of each target in each task.
- Some tasks were missed for both views: this test can be considered difficult, it may require more training.
- Successful targets were submitted on average at half of the deadline, which corresponds to the probable time taken for listening the first second of each sound sequentially.
Regarding stumble and recollection times:

- The number of stumble times differ from the number of success or recollection times: targets were stumbled upon but not submitted on a given task 8 times with the grid and 4 times with the cloud, subsequent tasks are required to see if a trend can be raised.

- From the recollection times, for both views, users take on average 10 seconds (slightly less for the cloud) to notice that the target has just been hovered and should be submitted, while each recollection median is at a few seconds; this elicits groups of reasons taking an increasing time: the interaction required to submit a sound (bumping the buzzer), proof-hearing the target again before submitting, or not noticing that the target was actually hovered (the latter probably in the last quartiles of times and outliers of each view).

![Stumble times in seconds](image1)

![Recollection times in seconds](image2)
Regarding distances and speeds:

- Distances are normalized to one row or column of 8 sounds in the square grid, thus the linear distance to listen to all sounds in the grid is around 10. Speed is computed by dividing distances by success times.

- Views for each task were scaled manually to maintain a constant area between tasks, however cloud view are more dense than grid views and their area seem smaller. This might explain why the first and third quartiles of distances are narrower for the cloud.

- The average speed of browsing for both views match: users seem to be willing to complete tasks with both views without favoring any by browsing faster with one of the two.

- The top outlier in cloud speed tasks is attributed to one of the creators of MediaCycle who was actually observed to be browsing quite faster during the tests.

Figure 5.17: Distances per view (in number of grid rows/columns) for test 1

Legend (box plot)
- interquartile range
- outliers (>3×IQR)
- outliers (>1.5×IQR)
- maximum
- quartile 3
- mean
- median
- quartile 1
- minimum
←→ tasks

Figure 5.18: Speeds per view (in grid rows/columns per second) for test 1

Legend (box plot)
- interquartile range
- outliers (>3×IQR)
- outliers (>1.5×IQR)
- maximum
- quartile 3
- mean
- median
- quartile 1
- minimum
←→ tasks
From the number of hovers and discovers, we can conclude that:

- The cloud required users to hover and discover more sounds to find targets, the analysis of the localization of targets and the spatial behaviors of search may provide a clue why: the bird-flight distances between the initial position of the pointer and the target are greater for cloud tasks, as seen on table 5.2.3.3.5.

<table>
<thead>
<tr>
<th></th>
<th>View</th>
<th>Grid</th>
<th>Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>0.84</td>
<td>1.22</td>
<td></td>
</tr>
<tr>
<td>Task 2</td>
<td>1.29</td>
<td>1.21</td>
<td></td>
</tr>
<tr>
<td>Task 3</td>
<td>0.82</td>
<td>1.41</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.99</td>
<td>1.28</td>
<td></td>
</tr>
<tr>
<td>Deviation</td>
<td>0.13</td>
<td>0.06</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5: Bird-flight distances to the target from the initial pointer position for each task for test 1

Figure 5.19: Discovers per view (number of unique sounds browsed per task) for test 1

Legend (box plot)
- IQR: interquartile range
- O: outlier (>3×IQR)
- +: outlier (>1.5×IQR)
- -: maximum
- ---: quartile 3
- ◊: mean
- —: median
- —: quartile 1
- -: minimum
←→: tasks

Figure 5.20: Hovers per view (number of cumulated sounds browsed per task) for test 1

Legend (box plot)
- IQR: interquartile range
- O: outlier (>3×IQR)
- +: outlier (>1.5×IQR)
- -: maximum
- ---: quartile 3
- ◊: mean
- —: median
- —: quartile 1
- -: minimum
←→: tasks
Table 5.6 reports the results of two-sample t-tests of the null hypothesis of equal means of these variables between both views.

<table>
<thead>
<tr>
<th>Variable</th>
<th>p-value</th>
<th>T</th>
<th>DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success times</td>
<td>0.70652</td>
<td>-0.37789</td>
<td>80</td>
</tr>
<tr>
<td>Stumble times</td>
<td>0.81300</td>
<td>-0.23724</td>
<td>92</td>
</tr>
<tr>
<td>Recollection times</td>
<td>0.48141</td>
<td>0.70734</td>
<td>80</td>
</tr>
<tr>
<td>Distances</td>
<td>0.72774</td>
<td>0.34936</td>
<td>80</td>
</tr>
<tr>
<td>Speeds</td>
<td>0.90831</td>
<td>-0.11554</td>
<td>80</td>
</tr>
<tr>
<td>Discovers</td>
<td>0.15918</td>
<td>-1.4211</td>
<td>80</td>
</tr>
<tr>
<td>Hovers</td>
<td>0.12439</td>
<td>-1.5529</td>
<td>80</td>
</tr>
</tbody>
</table>

From this table, we can notice that:

- Most p-values are quite higher than 0.05, showing that the results per variable are not drastically different between views, except discovers and hovers both get lower for the grid.

- The sign of most of the Student statistics favors the grid, except for recollection times and distances; this should be kept in mind for later test campaigns.

After the general comparison between views, we can analyze the trends for each task. Let’s first look at the top-ten shortest successful mouse paths of each task and target in Figure 5.21. First of all, the visual organizations induced patterns of spatial browsing, these strategies can be deciphered from the whole picture and discussed.

- For the grid, the shortest paths seem to have been more serendipitous than strategical: straight oblique ways, or sudden rectangular u-turns. Past these random chances, the “land mowing path” seems to be the fastest and most efficient way and was naturally chosen by most users: starting from the top left corner, line by line, but with a difference to western text reading in that audio lines were read alternatively forwards and backwards. Fewer users used a transposed version, mowing the map vertically, thus progressing in columns.

- The cloud was browsed more in a brushed or broomed way. A good strategy seems to be either looping around the periphery or diving straight inside, in both cases with an added oscillation. Users are and were more likely to intersect their previous ways with the cloud.

In the analysis of the second experiment of her PhD thesis 43, Kerry Rodden uses the terms systematic to describe a browsing approach such as the “land mower’s path” and haphazard for more “sketchy” pathways.

---

Figure 5.21: Top-ten shortest successful mouse paths for each task of test 1
Other considerations were observed during some tasks.

- The targets of the two first cloud tasks were spatial outliers in the view, these are circled in green in Figure 5.21. Most users avoided them. In contrast, this first grid task had a target located close in terms of “lawn mowing” distance.

- Some sounds had a long fade in time. Some users hovering on these too quickly would hear no sound and thus be troubled, questioning whether it was a bug from the user interface.

- Since the pointer was reset near the countdown column at the beginning of each task, some testers understood they had to move the pointer back to that position to re-hear the target sound, associating coordinates to any aural feedback, since all other sounds have a visual representation. This could be dealt with in subsequent tests by clearly explaining each tester they just need to untouch the trackpad and press the buzzer side button to re-hear the target.

- Through its regular structure inducing pathways for browsing, the grid compensates visual feedback on sounds already hovered.
Figure 5.23: Stumble times per grid then *cloud* tasks for test 1

Figure 5.24: Recollection times per grid then *cloud* tasks for test 1
Figure 5.25: Discovers per grid then cloud tasks for test 1

Figure 5.26: Hovers per grid then cloud tasks for test 1
**Experiment Evaluation**

**Distances in cells**

![Box plots showing distances in cells for different objects: ball (17/19), glass (11/19), scrape (17/19), bell (16/19), metal (11/19), water (10/19). Each box plot includes data points indicating variability.](image)

**Figure 5.27: Distances per grid then cloud tasks for test 1**

**Speeds in cells per second**

![Box plots showing speeds in cells per second for different objects: ball (17/19), glass (11/19), scrape (17/19), bell (16/19), metal (11/19), water (10/19). Each box plot includes data points indicating variability.](image)

**Figure 5.28: Speeds per grid then cloud tasks for test 1**
5.2.3.4 Experiment 2: content-based glyphs

Content-based positioning of sounds on a map being not efficient, we investigated whether adding content-based glyphs to represent sounds would improve reviewing search results.

This test was facilitated by Nicolas Riche at UMONS during practical work sessions in digital signal processing.

Table 5.7: Summary of experimental conditions for test 2

<table>
<thead>
<tr>
<th>Feature</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glyphs</td>
<td>yes</td>
</tr>
<tr>
<td>Collection(s)</td>
<td>1</td>
</tr>
<tr>
<td>Sounds (filtered)</td>
<td>by subset</td>
</tr>
<tr>
<td>Sounds (150)</td>
<td>150 sounds</td>
</tr>
<tr>
<td>Task deadline</td>
<td>60 s</td>
</tr>
<tr>
<td>Target choice</td>
<td>random</td>
</tr>
<tr>
<td>Tasks per user</td>
<td>10</td>
</tr>
<tr>
<td>Tasks per view</td>
<td>5</td>
</tr>
<tr>
<td>View sequence</td>
<td>interleaved</td>
</tr>
<tr>
<td>Testers</td>
<td>16</td>
</tr>
<tr>
<td>Recruiting</td>
<td>DSP students</td>
</tr>
<tr>
<td>Questionnaire</td>
<td>yes</td>
</tr>
</tbody>
</table>

5.2.3.4.1 System

Ware’s book offer great explanations and recommendations to use visual variables to support information visualization tailored for human perception. Grill et al.’s approach was to map many perceptual audio features to many visual variables (position, color, texture, shape), in one-to-one mappings. They chose to fully exploit the visual space by tiling textures: items are not represented by a distinct glyph, rather by a textured region. By undertaking an online evaluation based on a repertory grid method asking testers to rate sounds by choosing continuous values on scales of perceived features, they reported brightness to be one of the most salient statistically and also the most correlated to one of the signal features available from YAAFE: perceptual sharpness.


In a first attempt to discriminate the contribution of information visualization versus media information retrieval in sound browsing, we opted here for a simpler mapping. In a next iteration of the previous system, we mapped the mean over time of perceptual sharpness to the Value in the Hue Saturation Value (HSV) space of the node color for each sound, normalized against all sounds in each collection. We used the temporal evolution of perceptual sharpness to define a clockwise contour of the nodes. To compute the positions, perceptual sharpness was also added to the features from the former iteration of the system, intuiting it would gather closer items that are similar visually from their glyph representation.

5.2.3.4.2 Participants

16 participants (1 female) of average age 21.9 (+/-2.2) years old were recruited from students in Engineering (Digital Signal Processing) during two afternoons of group work, volunteering to take the test as a break. 8 had corrected vision.

5.2.3.4.3 Design

This controlled laboratory experiment was setup in a small control room embedded in a larger room where other students were working in groups, but without mutual visibility and limited sound interference. After a short 4-task introductory training, each tester performed 10 tasks, each time-limited to 60 seconds, passing automatically to the next after the deadline.

A single collection of 150 elements was chosen from a subset of the OLPC collection: the Berklee Sampling Archive Volume 7: noises (mechanical and industrial). Tasks toggled between layouts: grid without glyphs and cloud with glyphs, in an interleaved order.

Targets were chosen randomly at runtime, however the random seeds weren’t initialized at startup, therefore task sequences were similar daily between tests since the application was restarted for each participant. A post-test questionnaire was submitted to participants to collect demographic data and qualitative feedback. No financial reward was provided, but chocolate was offered.
5.2.3.4.4 Results

![Histograms of scores per user with the grid and cloud views for test 2](image1.png)

![Quantile plots of scores per user with the grid and cloud views for test 2](image2.png)

We again compute the Shapiro-Wilk test statistic (W) and its p-value to evaluate the normality of distribution of results: neither grid (W=0.85, p<0.01) nor cloud (W=0.89, p=0.04) scores look normal. We will thus again use Mann-Whitney’s u-test for further comparisons.

Table 5.8 reports the results of two-sample u-tests of the null hypothesis of equal medians of these variables between views.
Table 5.8: Mann-Whitney u-tests \((grid > cloud)\) of all variables for test 2: p-value, statistic \((Z)\), means per view \((M_{\text{view}})\). Bold means better.

<table>
<thead>
<tr>
<th>Variable</th>
<th>p-value</th>
<th>Z</th>
<th>(M_{\text{grid}})</th>
<th>(M_{\text{cloud}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success time (s)</td>
<td>0.04</td>
<td>-1.78</td>
<td>33.94</td>
<td>40.21</td>
</tr>
<tr>
<td>Stumble time (s)</td>
<td>0.81</td>
<td>-0.24</td>
<td>19.59</td>
<td>18.95</td>
</tr>
<tr>
<td>Recollection time (s)</td>
<td>0.48</td>
<td>0.71</td>
<td>10.51</td>
<td>15.17</td>
</tr>
<tr>
<td>Distance (cells)</td>
<td>0.02</td>
<td>-1.98</td>
<td>4.82</td>
<td>6.14</td>
</tr>
<tr>
<td>Speed (cells per s)</td>
<td>0.19</td>
<td>-0.86</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>Discovers (sounds)</td>
<td>0.16</td>
<td>-1.42</td>
<td>58.16</td>
<td>69.84</td>
</tr>
<tr>
<td>Hovers (sounds)</td>
<td>0.12</td>
<td>-1.55</td>
<td>33.38</td>
<td>38.68</td>
</tr>
</tbody>
</table>

For the evaluation of results for all remaining experiments starting with the current one, we will only analyze metrics per layout. We leave out metrics per target and distance paths to ease the reader’s task.

This time cloud (with glyphs) was significantly slower than grid to help users reach the targets.

For the music information retrieval community this result may be considered negative since a simple baseline solution outperforms a complex system with a layout obtained from a recent dimension reduction technique, carefully chosen feature extraction, both evaluated algorithmically in previously mentioned references, and glyph representation aiming at supporting audition with vision from perceptual cues.
Stumble times (Figure 5.33) somehow follow the trend of success times, except that grid stumble times show a lower first quartile, probably associated to inlier targets. Recollection times (Figure 5.34) look similar between layouts.

Figure 5.33: Stumble times per view (seconds) for test 2

Legend (box plot)
IQR interquartile range
\(\circ\) outlier (>3×IQR)
+ outlier (>1.5×IQR)
- maximum
— quartile 3
○ mean
— median
— quartile 1
- minimum
←→ tasks

Figure 5.34: Recollection times per view (seconds) for test 2

Legend (box plot)
IQR interquartile range
\(\circ\) outlier (>3×IQR)
+ outlier (>1.5×IQR)
- maximum
— quartile 3
○ mean
— median
— quartile 1
- minimum
←→ tasks
Participants browsed both layouts at a similar speed (Figure 5.36), inclining us to claim that no layout seem to have been favored for instance by participants who would have guessed which results we expected.

**Figure 5.35:** Distances per view (in number of grid rows/columns) for test 2

Legend (box plot)
- IQR interquartile range
- O outlier (>3×IQR)
- + outlier (>1.5×IQR)
- maximum
- quartile 3
- mean
- median
- quartile 1
- minimum
←→ tasks

**Figure 5.36:** Speeds per view (in grid rows/columns per second) for test 2

Legend (box plot)
- IQR interquartile range
- O outlier (>3×IQR)
- + outlier (>1.5×IQR)
- maximum
- quartile 3
- mean
- median
- quartile 1
- minimum
←→ tasks
Discovers (Figure 5.37) and hovers (Figure 5.38) should be compared to the collection size: 150 sounds. Random average chance is therefore 75 sounds, a threshold that both layouts fall underneath.

---

**Figure 5.37:** Discovers per view (number of unique sounds browsed per task) for test 2

Legend (box plot):
- IQR (interquartile range)
- O: outlier (>3 × IQR)
- +: outlier (>1.5 × IQR)
- maximum
- quartile 3
- mean
- median
- quartile 1
- minimum

**Figure 5.38:** Hovers per view (number of cumulated sounds browsed per task) for test 2

Legend (box plot):
- IQR (interquartile range)
- O: outlier (>3 × IQR)
- +: outlier (>1.5 × IQR)
- maximum
- quartile 3
- mean
- median
- quartile 1
- minimum
Figures 5.39, 5.40 and 5.41 display results of qualitative metrics obtained through a post-test feedback questionnaire. Even if users succeeded better with grid, they felt that cloud was better organized, more pleasurable and efficient (Figure 5.39).

To understand whether participants would be accustomed to rapidly scan a grid layout, they were asked their most used file browser layout. Result: the file list. (Figure 5.40).

Microsoft Windows is the most used operating system among users (Figure 5.41).
5.2.3.5 Experiment 3: expert students & competition

From the negative results obtained from the previous experiment, we targeted a different population sample closer to expert users (in sound auditioning), doubled the task deadline, and instilled a competition mood by announcing before the test session a give away of a prize to the best overall score.

This test was facilitated by Willy Yvart at Université de Valenciennes et du Hainaut-Cambrésis (UVHC) during free time of students, in a small cafeteria.

Figure 5.42: Setup of the third known-item search evaluation of sound layouts.
– © Willy Yvart

glyphs: yes
collection(s): 1
filtered by subset
sounds: 150 sounds
task deadline: 120 s
target choice: random
tasks per user: 10
tasks per view: 5
view sequence: interleaved
testers: 27
recruiting: AV students
questionnaire: yes

Table 5.9: Summary of experimental conditions for test 3

5.2.3.5.1 System

The system from the previous experiment was employed.

5.2.3.5.2 Participants

27 participants (6 female) of average age 21.3 (+/-2.2) years old were recruited from students in Audiovisual Communication, during two days. 13 had corrected vision. All the participants have studied audiovisual communication practices such as sound design and film edition.

5.2.3.5.3 Design

We slightly adapted our previous test design. We doubled the time deadline from to 120s.
We modified the test application to display a realtime score below the time countdown, computed as follows: $\sum_i t_{\text{max}} - t_i - \sum p_r - \sum p_f$ where $i$ is the number of tasks, $t_{\text{max}}$ is the time limit, $t_i$ the time of target finding at task $i$, $p_r$ the number of target rehears commanded by users, $p_f$ the number of false submissions. A prize in cash was awarded to the contestant with the best score, announced a few days before the tests, and proportional to the number of participants in order to invite them to call for challengers.

5.2.3.5.4 Results

![Histograms of scores per user with the grid and cloud views for test 3](image1)

![Quantile plots of scores per user with the grid and cloud views for test 3](image2)
We again compute the Shapiro-Wilk test statistic (W) and its p-value to evaluate the normality of distribution of results: this time both grid ($W=0.97$, $p=0.49$) and cloud ($W=0.92$, $p=0.03$) scores look normal. We will thus use Student-t tests for further comparisons, unpaired due to the existence of failed tasks. Table 5.10 reports the results of two-sample t-tests of the null hypothesis of equal means of these variables between both views.

<table>
<thead>
<tr>
<th>Variable</th>
<th>p-value</th>
<th>T</th>
<th>$M_{grid}$</th>
<th>$M_{cloud}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success time (s)</td>
<td>0.02</td>
<td>-2.04</td>
<td>50.18</td>
<td>56.29</td>
</tr>
<tr>
<td>Distance (cells)</td>
<td>0.06</td>
<td>-1.54</td>
<td>7.89</td>
<td>7.84</td>
</tr>
<tr>
<td>Speed (cells per s)</td>
<td>0.94</td>
<td>1.54</td>
<td>0.15</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Even with this preferred population sample closer to experts, cloud (with glyphs) still remains significantly slower than grid.

Figure 5.45: Success times per view (seconds) for test 3

Legend (box plot)
- IQR interquartile range
- O outlier ($>3\times$IQR)
- + outlier ($>1.5\times$IQR)
- maximum
- quartile 3
- ◇ mean
- red median
- ◇ quartile 1
- minimum
- ←→ tasks
Stumble times (Figure 5.46) somehow follow again the trend of success times, tighter than with the previous experiment. Recollection times (Figure 5.47) look similar between layouts.
Again speeds were similar between layouts. Testers in the current experiment were slower in absolute time but faster in relative time (to the task deadline) than testers in the previous experiments.
Discovers (Figure 5.50) and hovers (Figure 5.51) should be compared to the collection size: 150 sounds. Random average chance is therefore 75 sounds, a threshold that both layouts fall underneath. Surprisingly these amounts are greater than with the previous experiment. This may be due from the fact that the task deadline was doubled.

![Diagram of Discovers (sounds)](image1)

![Diagram of Hovers (sounds)](image2)
Figures 5.52, 5.53 and 5.54 display results of qualitative metrics obtained through a post-test feedback questionnaire. Even if users succeeded better with grid, they felt that cloud was better organized, more pleasurable and efficient (Figure 5.52).

The most used file browser layout is a file list (Figure 5.53).

Microsoft Windows is again the most used operating system among users, while here more participants are acquainted with Apple OSX, what is expectable from such a creative segment of the population (Figure 5.54).
Feedback was collected from the participants:

- Tester 1 said the *cloud view* reminded him of a “flock of sheep”, from the scattered positions.

- Tester 6 wished that “the trackpad was larger”, so that one stroke would cover the whole screen without breaking the path into relative chunks according to the trackpad dimensions. This could be solved by accessing directly to the absolute positions of each finger from a lower-level access than the standard desktop mouse support.

- Tester 12 preferred the term “English landscape garden” due to the dissymmetrical nature of the view.

- Tester 24 called the node glyphs “symbols”.

- Several users complained that some sounds were overlapping in the *cloud* layout. This must be addressed in following iterations, by adjusting the node sizes or positions.

- Several testers verbalized target sounds aloud at their first hearings, either by mimicking the sounds or by describing them mostly with the name of the source that seemed to have generated them. Some testers even used gestures to describe the sounds.
5.2.3.6 **Experiment 4: the metro layout**

From results obtained with the previous experiments, we posited that a layout with regular geometry such as the grid directs the search pathway and helps user keep a visual track of their progress in screening collections. We iterated our system with an overlap-free layout combining such practicalities of the grid and local similarity neighborhoods of the cloud.

![Figure 5.55: Setup of the fourth known-item search evaluation of sound layouts.](image)

**System**

We used the AudioMetro prototype described in the previous chapter (see 4.2.10). The major system update in comparison with the 2 previous experiments is that we introduced the metro layout that combines the benefits of a cloud layout and a grid layout.

**Apparatus**

The tests were undertaken on an Apple Macbook Pro Late 2013 laptop with 15-inch Retina display and resolution of 3360×2100, with a RME FireFace UCX sound card, and the same pair of Genelec active loudspeakers.
5.2.3.6.3 **Participants**

16 participants (5 female) of mean age 28 (+/- 6.3) each performed 9 tasks on 3 different collections. Besides 2 subjects, all the participants have studied or taught audiovisual communication practices (sound design, film edition). They were asked which human sense they favored in their work (if not, daily) on a 5-point Likert scale, 1 for audition to 5 for vision: on average 3.56 (+/- 0.60). All self-rated themselves with normal audition, 10 with corrected vision. Their self-rated average musical ear is 3.12/5 (+/- 0.31) in a 5-point Likert scale.

5.2.3.6.4 **Design**

We prepared 3 collections filtered by tag from the whole OLPC dataset: *water* (77 sounds), *spring* (93) and *metal* (147). An additional smaller collection was used for training tasks with each layout.

We qualitatively selected the optimal grid resolution based on the amounts of horizontal / vertical / diagonal adjacent neighbors computed for each resolution between the minimal side and the least distorted approximate, comparing such amounts between a proximity grid applied after dimension reduction and a grid ordered by filename. It is to be noted that not all collections obtained from other tags presented a proximity grid resolution that outperformed a simple grid by filename in terms of neighbor preservation.

Each layout was given a nickname: grid for the simple grid ordered by filename, album for its upgrade with glyphs, metro for the proximity grid of optimal resolution for neighbors preservation. These short nicknames brought two advantages: facilitating their instant recognition when announced by the test observer at the beginning of each task, and suggesting search patterns: horizontal land mowing for grid and album, adjacent cell browsing for metro. The metro layout was described to users using the metaphor of metro maps: items (stations) can form (connect) local neighborhoods and remote “friends” (through metro lines usually identified by color). Figure 5.56 illustrates all these layouts.
We sequenced the tasks for each tester as follows: *water metro, water album, water grid, spring grid, spring metro, spring album, metal album, metal grid, metal metro*. All collections exhibited several local neighborhoods with at least 3 very similar sounds that would end up close one another on each layout, exactly at the same positions between grid and album, elsewhere for metro. For each given collection, each layout was assigned one of such sounds as target.

We tamed the stress of users by removing the deadline and countdown display, only showing the score.
5.2.3.6.5 Results

We again compute the Shapiro-Wilk test statistic (W) and its p-value to evaluate the normality of distribution of results: this time both grid ($W=0.93$, $p=0.23$) and metro ($W=0.93$, $p=0.22$) scores look normal, album ($W=0.84$, $p=0.01$) scores don’t.

Thus, instead of ANOVA, we use the Kruskal-Wallis rank sum test (chi-square = 5.26 with $p=0.07$) which shows that there is almost a significant effect of layouts.

A Tukey multiple comparisons of success times means at a 95% family-wise confidence level on layouts shows that metro significantly outperforms grid ($p=.01$), while other comparisons are not significant: album is better than grid ($p=.34$) and worse than metro ($p=.26$).
This time all the tasks were successfully completed except two grid tasks (Figure 5.60).

Stumble times (Figure 5.61) follow again the trend of success times.
Recollection times (Figure 5.62) look similar between layouts, with outliers decreasing with the “content-basedness” of the layout.

![Recollection times in seconds](image)

Figure 5.62: Recollection times per view (seconds) for test 4

Legend (box plot)
- IQR interquartile range
- O outlier (>3×IQR)
- + outlier (>1.5×IQR)
- maximum
- quartile 3
- mean
- median
- quartile 1
- minimum
←→ tasks

Surprisingly speeds look lower for metro. This can be explained that metro layouts were more condensed (Figure 5.57), what has probably not been well compensated in the distance computation.

![Speeds in number of grid rows/columns per second](image)

Figure 5.63: Speeds per view (in grid rows/columns per second) for test 4

Legend (box plot)
- IQR interquartile range
- O outlier (>3×IQR)
- + outlier (>1.5×IQR)
- maximum
- quartile 3
- mean
- median
- quartile 1
- minimum
←→ tasks
Distances browsed are as well lower for *metro*. By looking again at the tasks (Figure 5.57), targets were located closer to the starting point for *metro* tasks.

![Distances in number of grid rows/columns](image)

**Figure 5.64**: Distances per view (in number of grid rows/columns) for test 4

---

Discovers (Figure 5.65) and hovers (Figure 5.66) are this time harder to compare due to the varying collection sizes (77-93-147 sounds) between triplets of tasks.

![Discovers (sounds)](image)

**Figure 5.65**: Discovers per view (number of unique sounds browsed per task) for test 4

---

Legend (box plot)
- IQR interquartile range
- O outlier (>3×IQR)
- + outlier (>1.5×IQR)
- maximum
- quartile 3
- mean
- median
- quartile 1
- minimum
- ←→ tasks
Figures 5.67, 5.68 and 5.69 display results of qualitative metrics obtained through a post-test feedback questionnaire.

Users felt that metro was more pleasurable and efficient, also better organized just behind album, agreeing with their performance (Figure 5.67).
The most used file browser layout is this time an icon grid (Figure 5.68), close to album and metro.

Microsoft Windows is again the most used operating system among users, and again participants are acquainted with Apple OSX (still a creative segment of the population) (Figure 5.69).

5.2.3.6 Discussion

These positive results open a promising track of investigation with the metro layout.

Feature extraction is a one-shot offline process at indexing time. Dimension reduction for layout computation is a process that should be close to real-time so as not to slow down search tasks and that is likely to be performed at least once per query. Decent results can be achieved by combining only content-based icons and simple ordering by filename. A content-based layout comes at a greater computational cost but brings significant improvements.
5.3 Conclusion

A contextual inquiry with sound designers convinced us to investigate solutions to facilitate sound browsing. Participating to the 2013 edition of the Video Browser Showdown with our video browser VideoCycle introduced us to known-item search tasks. We applied this method through four iterations of a usability evaluation, leading us to design a method to assist sound designers in reviewing results of queries by browsing a sound map optimized for nearest neighbors preservation in adjacent cells of a proximity grid, with content-based features cued through glyph-based representations. We showed that this solution was more efficient and pleasurable than a grid of sounds ordered by filenames.

Table 5.12 summarizes the experimental conditions that varied over the four experiments we described in this chapter.

<table>
<thead>
<tr>
<th>experiment</th>
<th>1 (pilot)</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>collection(s)</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>filter</td>
<td>tag</td>
<td>subset</td>
<td>subset</td>
<td>tag</td>
</tr>
<tr>
<td>sounds</td>
<td>64</td>
<td>150</td>
<td>150</td>
<td>77-93-147</td>
</tr>
<tr>
<td>task deadline (s)</td>
<td>60</td>
<td>60</td>
<td>120</td>
<td>unlimited</td>
</tr>
<tr>
<td>target choice</td>
<td>spread</td>
<td>random</td>
<td>clusters</td>
<td></td>
</tr>
<tr>
<td>glyphs</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>layouts</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>tasks per layout</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>layout sequence</td>
<td>interleaved</td>
<td></td>
<td>permuted</td>
<td></td>
</tr>
<tr>
<td>tasks per user</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>testers</td>
<td>19</td>
<td>16</td>
<td>27</td>
<td>16</td>
</tr>
<tr>
<td>recruiting</td>
<td>lab</td>
<td>DSP</td>
<td>AV</td>
<td>AV</td>
</tr>
<tr>
<td>questionnaire</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

(DSP: Digital Signal Processing, AV: AudioVisual)
6 Conclusion

SPACE: THE FINAL FRONTIER. THESE ARE THE VOYAGES OF THE STARSHIP ENTERPRISE. ITS 5 YEAR MISSION. TO EXPLORE STRANGE NEW WORLDS. TO SEEK OUT NEW LIFE AND NEW CIVILIZATIONS. TO BOLDLY GO WHERE NO [HU]MAN HAS GONE BEFORE.

STAR TREK (1966-PRESENT)

In chapter 2, we positioned this thesis in its context: improving the task of media browsing to support creative practices, aided by human-computer interaction and multimedia information retrieval. We explained some key terms that are recurrent in related works: media, content-based organization, similarity and navigation/browsing. We clarified the concerns in browsing media collections, ours being: blending techniques from human-computer interaction and multimedia information retrieval, evaluating browser prototypes with users to come up with better designs. We delimited the scope of our work by the issues it aims at solving: audio and video as media types, content-based rather than semantic organization, and visual display and gestural interaction as modalities.

In chapter 2, we investigated more into detail over research works that constitute the background of our own work: an overview on systems for browsing media content, from file browsers to media browsers (audio and video, sometimes for textual content), with content-based or semantic organization. We compared all these browsers along the media type they support, their type of organization, their input/output modalities (including dimensions of display and gestural interaction), the presence of user evaluations. We observed that most lacked a proper quantitative user evaluation and that browsing audio content in the context of sound design was less addressed than browsing music collections.

In chapter 3, we explained the method we chose to assist us in solving our research questions: combining interaction techniques (gestural input and information visualization) with a content-based organization workflow, illustrating it by reference works.
In chapter 4, we described our prototyping environment: the MediaCycle framework enabling applications featuring content-based organization, the DeviceCycle PureData toolbox for prototyping gestural interaction, paced by 3-month projects inside the numediart research program. We also presented a selection of the various media browsers it enabled.

In chapter 5, we analyzed our experimental results from the evaluation of some of our media browsers with users. One contextual enquiry with 6 sound designers helped us to better understand their practices and narrow down the scope of our research towards real issues. Our participation to the Video Browser Showdown live evaluation of video browsers got us acquainted with evaluation through know-item search tasks. We applied this evaluation method on 4 iterative experiments that lead us to improve the interactive presentation of search results on collections of sounds, combining advantages from both content-based organization and human-computer interaction.

In this closing chapter, we summarize our research contributions (section 6.1) and their limitations (section 6.2). We close up by providing clues for future works (section 6.3).

6.1 Summary of research contributions

Our doctoral work aimed at improving the user experience of browsers of media collection organized by content-based similarity and influenced by design cues from information visualization and gestural interaction.

We produced an in-depth state-of-the-art on media browsers (for audio and video files, organized by content-based similarity or semantically) that revealed that browsers for sound collections have been gaining less research interest than browsers for music libraries.

We first showed through user evaluations by known-item search tasks in collections of textural sounds that a baseline grid layout ordered by filename unexpectedly outperforms a content-based similarity layout resulting from a recent dimension reduction technique (Student t-distributed Stochastic Neighbor Embedding) applied to a feature set, even when complemented with content-based glyphs that emphasize local neighborhoods. Researchers usually take for granted that complex content-based algorithms are assumed to be more efficient than simpler baseline solutions.

One design cue raised from observations of users taking the experiments and the quantitative analysis of their logs is that a grid layout directs the search paths and helps users keep track of elements already browsed. We proposed a solution informed by these findings and that elegantly mixes MIR and HCI techniques: AudioMetro. After dimension reduction, we applied a proximity grid (a solution borrowed from image browsing), optimized to be the densest possible while preserving nearest neighborhoods. Our last user evaluation revealed promising results in favor of our design, explained as an analogy to a metro map with similarity conveyed as metro lines between stations.
6.2 Discussion

In this section we focus our discussion on the evaluation protocol we employed. In the next section opening towards future works (6.3), we’ll discuss other aspects that can be addressed regarding the design of media browsing systems.

6.2.1 Evaluating research prototypes versus real-life tools

The systems we built were abstractions of a real-life tools evaluated through controlled lab experiments. Our work should be integrated in a real-life system and evaluated through more complex tasks (for instance “create a sound fitting the following textual description”), and through a longer-term incubation into the daily practices of such users. At least we took care to recruit testers as close as possible to the expected expert users of the tools we modeled: students in audiovisual communication (some would-be sound designers).

6.2.2 Designing the experimental protocol

Even if it is a well documented issue we were aware of, we have understood through experience that experimental variables and settings highly influence results. This is what researchers replicating the evaluation of the Great CHI’97 Browse-Off forum faced, some of the experiments invalidating the contest results\(^1\). Would we have obtained positive results in our fourth experiment with different variables such as population sample, task order and amount, sound collections?

6.2.3 Experimental sample size and online evaluation

The low amount of users and tasks in our last experiment questions its statistical significance. Porting our system to a web-based application may help to reach more testers, but with the caveat of not being able to monitor the experiments to ensure testers remain focused on the tasks.

6.2.4 Exploring content-to-visual variables mappings

We scratched the surface in terms of content-to-visual variables mappings: mostly position, followed by color and contour. It was a conscious choice so as to introduce the fewest experimental variables into the tests. A deeper exploration of such mappings is required as follow-up work.

---


6.3 Future works

6.3.1 Direct improvements in our research tracks

Future tracks can be addressed by the MIR and HCI research communities.

6.3.1.1 MIR-related

Improving our work would require to investigate all blocks from the multimedia information retrieval data flow: feature extraction, dimension reduction and layout computation.

6.3.1.1.1 Descriptors for sound effects

Other features tailored for sound effects should be tried. The Essentia framework provides in a “sfx” category some descriptors such as: the logarithm of the attack time for the sound, temporal locations of the maximum or minimum amplitude values starting from the beginning or end of the sound, the normalized position of the temporal centroid 3.

6.3.1.2 Dimension reduction

Figure 6.1 shows a tSNE-based cloud layout of a collection of sounds filtered by tag “water” from the OLPC library, without (greater clarity) and with high-dimensional nearest-neighbor links: each element is linked to its first nearest neighbor in the full feature set dimension, the thicker the link is the greater the pairwise similarity is. Two such links numbered 1 and 2 on the Figure are thick and large: the pairs of elements linked are very similar in high-dimension (with glyphs looking very similar) but very far in 2D. Reducing pairwise distance preservation errors may be an investigation track.

---


---

Figure 6.1: tSNE-based cloud layout of OLPC tag-based collection “water” without (left) and with (right) high-dimensional nearest-neighbor links
6.3.1.1.3 Proximity grid optimization To prepare each sound collection for the AudioMetro evaluation, we optimized the proximity grid by picking manually its best grid side for neighborhood preservation. Automating this process would remove human supervision, as in very recent approaches such as MaxSAT by Bunte et al.\(^4\) and Self-Sorting Map by Strong and Gong\(^5\). Can a radically new dimension reduction and grid algorithm be developed so as to maximize the criterion that high-dimensional nearest neighbors should remain close in 2D?

6.3.1.2 HCI-related

6.3.1.2.1 Visualizing similarity in 1D with lists Most commercial applications for media asset management use list or spreadsheet views, allowing different ways of sorting per categories of metadata. Conveying similarity through 1D, rather than 2D as in the current work, may help to adapt such systems directly.

6.3.1.2.2 Combining tag- and content-based views A subsequent iteration of this system should be designed to synchronize a content-based map representation with a context-based tag view. Tag clouds are popular but not necessarily the most efficient layout\(^6\). Such an hybrid system would allow filtering by tag and browsing by similarity in the same application.

6.3.1.2.3 Designing suitable gestural interaction The most obvious reason that we used an optimized proximity grid was to combine the benefits observed through user evaluations of a content-based cloud layout (similarity neighborhoods) and a grid layout (directing search pathways). The meshed nature of grid layouts could also facilitate the design of suitable gestural interaction to augment the search experience, for instance through sound-informed force-feedback physical hover when browsing sound collections as explored with our Starfish eNTERFACE (see 4.2.9). Discrete displays and devices are easier to manufacture when they feature grid patterns: LED displays, capacitive touch films... Beyond flat displays meshed in pixels, we want to investigate the use of tangible interfaces, such as the "post-pixel" tangible interfaces of Follmer et al.\(^7\).


6.3.2 Extended research tracks to explore

Shneiderman defines the 4 genex\(^8\) phases of creativity \(^9\):

- **collect**: searching and browsing digital libraries
- **relate**: discussing new ideas with others
- **create**: exploring solutions
- **donate**: disseminating results

We can consider that our work mainly contributed to the first genex (**collect**). Outside the scope of our research, the integration of our work into cloud-based database-powered systems (**relate**), authoring systems (**create**) and archival systems (**donate**) can be imagined.

---

\(^8\) for generator of excellence, as homage to Vanevar Bush’s *memex* for memory extender

7 References

If you’re a researcher on this book thing and you were on earth, you must have been gathering material on it.

Douglas Adams ¹

7.1 Books


7.2 Journal articles


Schoeffmann, Klaus, Hopfgartner, Frank, Marques, Oge, Boeszoermenyi, Laszlo, and Jose, Joemon M. “Video browsing interfaces and applications: a review”. In: SPIE Reviews 1.1, 018004 (2010), pp. 1–35. DOI: 10.1117/6.0000005 (cit. on p. 103).


7.3 Conference proceedings


Bärecke, Thomas, Kijak, Ewa, Nürnberg, Andreas, and Detyniecki, Marcin. “Video navigation based on self-organizing maps”. In: Proceedings of the 5th International Conference on
References


-- “Nonlinear dimensionality reduction approaches applied to music and textural sounds”. In: IEEE International Conference on Multimedia and Expo (ICME). 2013 (cit. on pp. 111, 204).


Frisson, Christian. “Conception centrée utilisateur de prototypes interactifs pour la gestion de contenu multimedia par similarité”. In: Rencontres doctorales de la 22ème Conférence


Frisson, Christian, Dupont, Stéphane, Siebert, Xavier, Tardieu, Damien, Dutoit, Thierry, and Macq, Benoît. “DeviceCycle: rapid and reusable prototyping of gestural interfaces,
applied to audio browsing by similarity”. In: *Proceedings of the New Interfaces for Musical Expression++ (NIME++)*. Sydney, Australia, 2010 (cit. on pp. 37, 156, 159).


Julià, Carles F. and Jordà, S. “SongExplorer: a tabletop application for exploring large collections of songs”. In: 10th International Society for Music Information Retrieval Conference. 2009 (cit. on p. 75).


Lawson, Jean-Yves Lionel, Al-Akkad, Ahmad-Amr, Vanderdonckt, Jean, and Macq, Benoit. “An open source workbench for prototyping multimodal interactions based on off-the-


Rice, Stephen V. “Frequency-Based Coloring of the Waveform Display to Facilitate Audio Editing and Retrieval”. In: 119th Convention of the Audio Engineering Society. New York, New York USA, 2005 (cit. on p. 120).


Song, Yaxiao, Marchionini, Gary, and Oh, Chi Young. “What are the most eye-catching and ear-catching features in the video?: implications for video summarization”. In: Pro-


7.4 Reports


Peeters, G. A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project. Tech. rep. IRCAM, 2004 (cit. on p. 109).

7.5 Master and doctoral theses


Designing interaction for browsing media collections (by similarity)

Sound designers source sounds in massive and heavily tagged collections. When searching for media content, once queries are filtered by keywords, hundreds of items need to be reviewed. How can we present these results efficiently? This doctoral work aims at improving the usability of browsers of media collections by blending techniques from multimedia information retrieval (MIR) and human-computer interaction (HCI). We produced an in-depth state-of-the-art on media browsers. We overviewed HCI and MIR techniques that support our work: organization by content-based similarity (MIR), information visualization and gestural interaction (HCI). We developed the MediaCycle framework for organization by content-based similarity and the DeviceCycle toolbox for rapid prototyping of gestural interaction, both facilitated the design of several media browsers. We evaluated the usability of some of our media browsers.

Our main contribution is AudioMetro, an interactive visualization of sound collections. Sounds are represented by content-based glyphs, mapping perceptual sharpness (audio) to brightness and contour (visual). These glyphs are positioned in a starfield display using Student t-distributed Stochastic Neighbor Embedding (t-SNE) for dimension reduction, then a proximity grid optimized for preserving direct neighbors. Known-item search evaluation shows that our technique significantly outperforms a grid of sounds represented by dots and ordered by filename.

Christian Frisson

Christian Frisson was born in Thionville, France in 1982. He graduated a MSc. in “Art, Science, Technology (AST)” from Institut National Polytechnique de Grenoble (INPG) and the Association for the Creation and Research on Expression Tools (ACROE), France, in 2006. In 2015, he obtained his PhD degree from the University of Mons (UMONS) on designing interaction for organizing media collections (by content-based similarity). He has been a fulltime contributor to the numediart Institute since 2008.

http://frisson.re

Université de Mons
20, Place du Parc, B7000 Mons - Belgique
Tél: +32(0)65 373111
Courriel: info.mons@umons.ac.be
www.umons.ac.be