Master Thesis

Exploring Complementarity in a Multi-Device Spatially Tracked Music Production

Master Thesis
Group IS103F16
Software
Department of Computer Science
Aalborg University
Spring 2016

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Exploring Complementarity in a Multi-Device Spatially Tracked Music Production

Project period:
P10, spring 2016

Project group:
IS103F16

Group members:
Christian Jødal O'Keeffe
Rasmus Fischer Gadensgaard

Supervisor:
Jesper Kjeldskov

Total number of pages:
41

Department of Computer Science
Selma Lagerlöfs Vej 300
DK-9220 Aalborg East
http://www.cs.aau.dk/en

Synopsis:

This report explores extension of an existing multi-device ecosystem. This is done through the implementation of a system for a specific case: Music production. The system, documented with a paper, is for mixing audio tracks using a computer, a physical control surface and an iPad. The extension consists of a spatially aware iPad, showing relevant information, and facilitating new interactions through the iPad. The location of the iPad can be used to interact with the other parts of the system.
A study of use of the system has been conducted by expert users from the music department at Aalborg University. They found that the interaction facilitated by the iPad was useful and brought them closer to the music production, however the spatial tracking was confusing. From the study, a set of guidelines has been made, which can be used for further research and development of multi-device extension systems. The report also describes some of the central elements of implementation, as well as ideas for future work.

Preface

This report is written as a master thesis by group IS103F16, which consists of Christian Jødal O'Keeffe and Rasmus Fischer Gadensgaard from the Department of Computer Science at Aalborg University. The report contains the documentation of the development and subsequent study of use of a piece of multi-device extension software used in the context of a music production.

The sources in this report are on the form [Author, Year], with a corresponding entry in the bibliography. The article included has a separate bibliography, with sources on the form [Reference number].

A project disc is included in Appendix A containing e.g. this report, source code as well as an appendix report with transcripts, video recordings and logs.

A video demonstration of the developed system can be seen at `https://goo.gl/GUGgtm`.

We would like to thank our supervisor Jesper Kjeldskov for guidance and feedback throughout the project. We would also like to thank assistant professor Mads Walther-Hansen for sharing his knowledge of music production, mixing and further music-related feedback and advice in the project as well as providing equipment and helping with designing the assignments for the evaluation. At last we would like to thank the participants for helping evaluation the product.

Aalborg, June 14, 2016

Christian Jødal O'Keeffe

Rasmus Fischer Gadensgaard

Contents

1 Resumé

Dette projekt omhandler udvidelsen af et eksisterende multi-enheds digitalt økosystem inden for området af musikproduktion. Projektet er lavet i samarbejde med musikafdelingen på Aalborg Universitet, som har givet vigtig viden og indsigt i området omkring musikproduktion.

Et digitalt økosystem i musikproduktion består typisk af mange forskellige enheder, som kommunikerer på tværs af hinanden. Et eksempel på sådan et økosystem kan for eksempel være en computer med tastatur og mus, en mixerpult og instrumenter. På computeren bruges programmet Logic Pro X ofte til at producere og mixe musik. Det er opbygget således, at man kan placere plugins på hvert lydspor, hvilket giver en bestemt effekt i lydudtrykket for den pågældende kanal.

I dette projekt er der blevet lavet en udvidelse af det eksisterende digitale økosystem inden for musikproduktion. Denne udvidelse består i at en spatial bevidst tablet er blevet tilføjet til økosystemet. Denne tablet kan bruges til at interagere med økosystemet ved hjælp af dens placering på bordet, hvor skærm og mixerpult står. På computeren kører programmet Logic Pro X, som bruges til at redigere og producere musik. En lydproduktion er normalt opbygget af forskellige lydkanaler, som skal mixes i forhold til hinanden for at få et tilfredsstillende stykke musik, bl.a. vha. plugins. Når en produktion har en vis størrelse, har man mange forskellige lydspor og mange forskellige plugins placeret på disse. Det kan derfor være svært at overskue de mange plugins på den begrænsede mængde skærmplads.

Tabletten er sporet vha. en Kinect, så man kan ved at flytte tabletten til venstre skifte den markerede kanal til venstre, og ligeledes ved at flytte tabletten til højre skifte kanal. Derudover viser tabletten de plugins som er tilføjet til den nuværende markerede kanal. Det er muligt at åbne et plugin på tabletten og interagere med dette via tablettens berøringsfølsomme skærm. Dette giver altså nye former for interaktioner med plugins. Pluginet vil så kun blive vist på tabletten, og dermed ikke fylde på computerskærmen. Pluginet bliver husket, så rykkes tabletten væk fra kanalen lukkes pluginet, indtil tabletten rykkes tilbage til kanalen igen, hvor det vil åbnes igen.

Systemet er blevet evalueret af musikstuderende ved Aalborg Universitet med omfattende kendskab til musikproduktion gennem Logic Pro X. Gennem denne evaluering kom det frem at deltagerne generelt godt kunne lide muligheden for interaktion med plugins via den berøringsfølsomme skærm. Ved at flytte plugins til en tablet, kom informationen desuden tættere på deltagerne, hvilket ifølge nogle deltagere gjorde, at de følte en øget samhørighed og nærvær med systemet. Den spatiale del af tabletten fandt deltagerne dog mere en distraktion frem for produktivitetsfremmende. Dette kan dog skyldes implementationen af denne funktionalitet, som ud fra evalueringen, synes at være ulogisk. Derudover blev det fundet at en udvidelse af et eksisterende system ikke nødvendigvis betyder, at interaktionen fra det eksisterende kan overføres direkte til andre typer af enheder.

Baseret på evalueringen, er fem forskellige guidelines blevet udviklet. Første guideline er, at indhold, input metoder og interaktioner skal passe til den specifikke enhed. Hver enhed har sine fordele og ulemper i forhold til præsentation af indhold og interaktion, og det bør der derfor tages hensyn til. Den anden guideline er, at formålet med en enhed bør matche dens funktionalitet, så en enhed med en meget lille skærm f.eks. ikke bliver brugt til at have det store overblik. Hver enhed bør have formål, som passer til enheden.

Tredje guideline er, at hvis en adfærd for systemet tilsidesættes eller ændres, bør dette fremgå meget klart af systemet, så man ikke er i tvivl om at den givne funktionalitet er ændret.

Fjerde guideline er, at synkroniseringen mellem enhederne i et økosystem bør være stabil, hurtig og samtidig, så brugeren oplever en klar forbindelse mellem enheder, og oplever at indhold og data hele tiden er opdateret i forhold til hinanden. Den femte guideline er, at systemet bør være robust og fri for fejl. Dette er specielt vigtigt i multi-enheds systemer. Disse guidelines kan bruges til at designe fremtidige multi-enheds digitale økosystemer, og kan ses som en udvidelse til Sørensen et al.'s eksisterende 4C rammeværk, som kan bruges til at beskrive multi-enheds digitale økosystemer.

Introduction 2

As personal digital ecosystems have developed and expanded to an increasing number and types of devices, new possibilities arise in terms of working digitally. These devices are all connected in the digital ecosystems and can communicate internally [Gable et al., 2015]. Since the graphical user interface was developed, a lot of possibilities arose, but also problems like e.g. the problem of having enough screen space for simultaneously open applications and windows has persisted. Attempts to mitigate this problem includes having multiple screens, larger screens and higher screen resolutions [Czerwinski et al., 2003; Reda et al., 2015; Seyed et al., 2013; Vatavu and Mancas, 2014]. This was followed by the rise of multi-device ecosystems. These additional devices can therefore be used to create a richer working experience as they can aid the working processes of the user. Devices are used together, both simultaneously and sequentially. To describe these multi-device systems, Sørensen et al. presents the 4C framework, which consists of four themes: Continuity, Collaboration, Communality and Complementarity [Sørensen et al., 2014]. These four themes, each with two underlying principles, describes the four cases of simultaneous vs. sequential use and many users vs. many artifacts. This framework is a powerful tool for exploring and describing multi-device systems.

There already exists much research on the area of cross-device interaction [Aumi et al., 2013; Hamilton and Wigdor, 2014; Pierce and Nichols, 2008; Rädle et al., 2014; Schmidt et al., 2012; Skov et al., 2015]. This paper however describes a special case of cross-device interaction within the theme Complementarity, with the case of music production. Music production is already today a multi-device ecosystem with computers, control surfaces etc. This can give a high degree of complexity when new devices are introduced in the ecosystem. History shows that music production including music mixing, has introduced several new devices over the years, but still keeping some of the old metaphors. An example is the increasing importance of the computer in music mixing, where more and more functions are moved away from e.g. control surfaces and other physical devices, so software on a computer like e.g. Logic Pro X includes metaphors for these functions. This introduction of the computer gave many new possibilities, and changes the working style of the producer. An introduction of another device could have a similar impact, or aid the producer with the work on mixing music together with the existing devices, improving their overall work and their use of the existing devices. One of the challenges in multi-device systems can be the understanding of the devices relations to each other. In relation to the music production case, it could be which channels are altered in the different devices, as a change on the physical control surface e.g. could be to another channel than the one marked in Logic Pro X. Other research has explored the use of spatial awareness to aid the user to understand the devices' relation [Nielsen et al., 2014; Rädle et al., 2014].

Inspired by the above, we will explore the theme of Complementarity, with the principle of extension, and seek to answer the research questions:

- RQ1: *How can a cross-device system be designed, which facilitate interaction during music production?*
- RQ2: *How can a spatially tracked multi-device system be developed to assist music production?*

Research Paper 3

To answer the research questions in Chapter 2, a multi-device system was developed, which was evaluated through a study of use. The following paper details the results from this study.

3.1 Paper: Exploring Complementarity in Multi-Device Spatially Tracked Systems

As more and more devices with different functionality are added to digital ecosystems, these sometimes support different types of interaction, which need to be kept in mind when designing multi-device systems. To describe types of multi-device systems, the 4C framework can be used. This project explores the 4C framework principle Extension within the theme Complementarity, with the case of music production. A system which enables music producers to move content onto a spatially tracked iPad has been developed, which serves as a proof-of-concept implementation to gain additional knowledge about the principle of extension. This system was evaluated by expert users through a study of use, where valuable knowledge about the use of devices in relation to each other was discovered.

The system developed in this project was developed as an extension to the existing way of producing music, where producers in the studied case use a computer with the program Logic Pro X. The extension consists of an iPad which is spatially tracked in relation to the physical control surface. This iPad can be used to change the active channel in Logic Pro X by moving the iPad physically. Users can view information about the active channel, such as which plugins are added to the channel. They can also open plugins directly on the iPad, and interact with them with the iPad's multi-touch interface.

A study of use was conducted by four expert participants of Logic Pro X, which revealed that the participants found the idea of being able to interact with plugins using their hands useful and interesting. The spatial awareness of the iPad was however regarded as confusing. In order to help future development of multi-device systems, a set of guidelines has been derived from the results of the study of use, which can be regarded as an extension to the 4C framework.

A video demonstration of the developed system can be seen at `https://goo.gl/GUGgtm`.

Exploring Complementarity in a Multi-Device Spatially Tracked Music Production

Christian Jødal O'Keeffe and Rasmus Fischer Gadensgaard
Department of Computer Science
Aalborg University, Denmark
{cjok11, rgaden11}@student.aau.dk

ABSTRACT
This paper investigates extension of an existing multi-device digital ecosystem within the area of music production. The extension consists of a spatial aware tablet, which show and enable interaction with relevant elements of the system according to its spatial location. The system can also be directly interacted with by moving the tablet left or right. The case used for this study is a system for music producers consisting of a physical control surface, iPad and a desktop computer with keyboard and mouse. A study of use of the system has been conducted, which revealed, that experienced music producers found the interaction on the tablet very useful, but the tracking of the tablet a distraction and counter productive. From the knowledge gained through the study of use, a set of guidelines for multi-device development has been formed, which serves as suggestions for future research and development of multi-device system and extension of these.

Author Keywords
Complementarity; Extension; Spatial Awareness; Music Production; Color Tracking; Multi-Device Interaction

INTRODUCTION

In this paper multi-device interaction in a music production environment is explored. Music productions often include many different types of devices such as computers, control surfaces, instruments etc. An example of a mixing studio for music production can be seen on Figure 1. These devices together form a multi-device ecosystem.

Figure 1: An example of a mixing studio. This is a multi-device ecosystem with many different devices. The mixers and control surfaces can be connected in multiple ways.

The system seen in Figure 1 is a multi-device ecosystem where the devices are used simultaneously, but the interaction of each device has however not been designed to facilitate the use of other devices in parallel.

Being able to keep an overview of a music production project is very important, as the producer must be able to fine tune many different elements of a of music production according to each other. As a way of accommodating the ability to have much information and alter this simultaneously, the idea of having a movable interactive screen, which show different information according to its spatial position, sparked. We chose to explore the use and interaction of a multi-device spatially aware system within music production. To research the area of multi-device interaction in music production, we have developed a prototype system as an extension to the existing music production ecosystem. A better knowledge of multi-device systems will result in more and better systems, making it possible for developers to use this knowledge to create systems, which will facilitate multi-device interactions better, and create a better user experience in the end.

The extension adds an additional device, a tablet, to the ecosystem which can be used to interact with other devices in the ecosystem and to display and interact with relevant information on its screen according to the tablets spatial location. The tablets location is spatially tracked in regard to the other devices in the system, in this case a physical control surface. This system has been evaluated in a study of use by expert users with musical background. The results showed that the users found the interaction on the tablet very useful, however the spatial awareness of the tablet was regarded as a distraction rather than useful.

Multi-device interaction can be described by the 4C framework presented by Sørensen et al. [10], and the aim of this paper is to supplement the framework with a higher empirical knowledge, extending the description of a multi-device ecosystem provided by the 4C framework. From the results, a set of guidelines for facilitating the theme of Complementarity, from the 4C framework, in multi-device ecosystems has been formed.

RELATED WORK

In the following we outline central related work of the large body of research within the field of multi-device interaction. This section has been grouped into six different themes of related work, including a section of related commercial products.

System Extension

One area of research of multi-device systems focus on the extension of existing systems. ActivitySpace, presented by Houben et al. [4], is an information management system, which enables resource management through an interactive desk. A device is placed on the desk, and the desk visualizes resources such as documents and media files around the device. These resources can be easily moved between the devices, by performing a drag and drop gesture on the desk. Activities such as writing a document can take place on the interactive desk, utilizing resources from the connected devices. This form of system extension enables the users to manage device ecosystems simply by placing them on a desk.

Similarly, Chen et al. present Duet [2], which is a cross-device system exploring interactions between smartwatches and smartphones. Using Duet it is possible for the two devices to have different roles according to how they are currently being used. For example, if the phone is being actively used, the smartwatch can simply act as a sensor, enabling new interactions, depending on the rotation of the watch, such as using the knuckles for interacting with the smartphone.

Interaction Techniques

There has been much research within interaction techniques facilitating cross-device interaction and research of how users interact with devices when presented a multi-device system. Rädle et al. [7] developed five different cross-device interaction techniques during their development of the HuddleLamp system. A simple techniques called *peephole navigation* was developed to demonstrate the spatial relationship of the tracked device. Here a map was displayed on the device, which could be navigated by moving or rotating the device physically. Multiple devices could be placed next to each other, thereby creating a larger view of the map. The system consists of a desk lamp with an integrated camera for tracking. The camera tracks the desk, where multiple devices are placed, and enables collaboration between the devices using a set of interaction techniques such as *Huddle navigation*, where a device is used to navigate a map. When two devices are moved close together, they connect and form a unified image.

Hamilton and Wigdor presented the Conductor framework [3], in which different devices can be arranged in symphonies in order to be perform tasks cross-device. The evaluation of the system showed, that participants made use of cross-device functionality when possible, in fact two participants of the evaluation used as many as 10 different devices simultaneously. The study showed that people are able to comprehend working with multiple devices simultaneously.

Spatial Awareness

A different field of multi-device systems are spatial awareness, where devices are aware of their position in relation to other devices. Spindler et al. [11] presented a low-budget system, which utilized a Kinect to spatially track objects such as iPads. Their research was a continuation of an earlier project, which used expensive equipment [12] and which therefore in their opinion was not applicable outside of research facilities. Their system is based on a collaborative line of thought, where a ceiling projector creates a collaborative space on a tabletop. This tabletop can be interacted with through two approaches: Active and projective. In the active approach users interact with the tabletop through active tangible devices such as tablets and smartphones. In the projective approach, pieces of paper and cardboard can be used as displays, as long as they are within reach of the projector, so that it can project an interactable interface onto the object. Their findings reveal that a Kinect is in fact able to track objects spatially. It however has problems in some cases, such as when objects are angled steeply towards the tracking devices, or when objects are far away.

JuxtaPinch [6] presented by Nielsen et al., is a system where multiple devices can be connected by performing a pinching gesture at the edge of the screen of each device. The devices connect, and together form a display, which can be used for showing images.

Multi-Device in Music Production

Within the area of music production and multi-device systems, Xambó et al. [13] conducted a long-term study of the interaction of the system Reactable, which is an interactive table designed for music creation through the use of tangible objects. The system contains different types of physical objects such as sound generators and sound effects. Based on the spatial location of these objects, the Reactable creates sound, according to both distance between objects and rotation of each object in relation to each other. The study showed that the design of the Reactable promoted self-regulation of the space, so that multiple users where able to collaborate in a joint music experience.

Martin et al. present a similar study with a multi-device music system, where iPads where used in a music ensemble [5]. In the study they discover, that the introduction of iPads as instrument made the musicians use new gestures such as fast and slow swirls, taps with a swipe and even gestures where both hands were used to indicate rhythm.

Themes of Multi-Device Use

Other research has focused on the conceptual parts of multi-device systems. Sørensen et al. [10] present a framework categorizing four distinct categories of multi-device interaction: Communality, Continuity, Collaboration and Complementarity, as seen in Figure 2.

The theme Complimentarity describes simultaneous interactions on many artifacts. The principle Extension refers to when multiple devices complement each other in providing more interaction possibilities. An example of this is the Microsoft Xbox "SmartGlass" technology, where players can connect their Android and iOS devices to the Microsoft Xbox, and use their device to get related information such as a game map while playing.

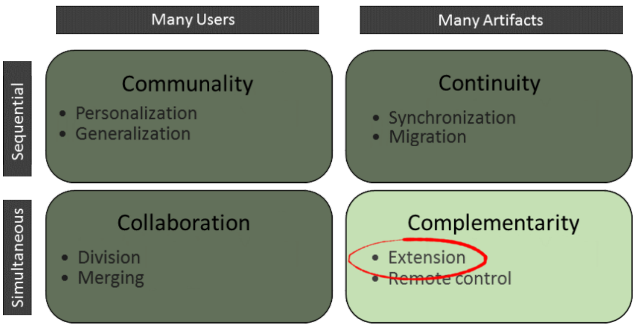


Figure 2: An overview of the 4C framework. Complementarity is highlighted because of the focus of this paper.

Commercial Products

There already exists applications which makes it possible to extend music production systems to an iPad. An example is Logic Remote, which extends Logic Pro X (a computer program for mixing and producing music) to an iPad. The purpose of this application is to have the full Logic Pro X interface presented on the iPad, but using different views for different areas of Logic Pro X. It is possible to open plugins on channels using Logic Remote, however the representation of some plugins are altered, for example the compressor plugin does not have the beat indicator as found in the desktop version.

Using remote control software, it is possible to extend most applications to a tablet, such as an iPad. Example of this type of software include Teamviewer, Splashtop, RealVNC and LogMeIn. This type of software makes it possible to remote control a desktop computer with other devices, such as tablets. Some of these applications also enable extending the desktop system to the screen of a tablet, making it possible to move program windows etc. to the tablet screen, and interact with the program on the tablet without displaying the window on the computer screen.

MUSIC PRODUCTION CASE

The case of this study is a mixing studio for music production. The case is chosen due to the fact, that this case is already a complex multi-device setup, as seen on Figure 1. In this multi-device system, the computer is the main device, on which the software for mixing runs. To interact with this software, several control surfaces are placed along with keyboard, mouse and other devices.

The system is thereby a multi-device mix of both old-fashioned and new devices, where the devices form a multi-device system, which can communicate, but was not designed for simultaneous use, and was not designed as a complete digital ecosystem. This makes the case a good case for exploring the effects from adding an additional device, which should not only work together with one, but several other devices, in an environment of habits and very specific needs, with the focus on extending the devices as a digital ecosystem where the devices' relation are important for the extension.

To understand the case, we collaborated with the music department at Aalborg University. We had several meetings with Mads Walther-Hansen, PhD, Assistant Professor, from which we got valuable knowledge about the case. We saw a mixing studio, and was introduced to how music producers mix a track and use the different devices and software. From this knowledge, we learned that producers usually have a set of raw sound files, from which they will mix a track by altering audio channels with effects. One widespread used software to do this is Logic Pro X, which can be seen on Figure 3. Here you load in the sound files to channels, and add effect by adding and adjusting a series of different plugins.

Figure 3: Logic Pro X with one plugin open. The plugin uses a lot of screen space, hiding important information underneeth.

Though the collaboration with the music department, it was found, that the computer screen can quickly become cluttered with plugins and windows while mixing a track with effects as seen in Figure 4. Many of the same plugins are used on many of the same channels, which cause confusion. At the same time, the plugins were a central part of their work with mixes. We saw therefore potential in enabling the system to support this use of plugins, which should work together with the existing systems, enabling the users to work in a different way due to the new device and its opportunities, e.g. reducing the clutter on the screen and enabling the interaction with the central plugins in a new way. Therefore mixing a track using effects by plugins was the starting point of the case to explore Complementarity.

Figure 4: Logic Pro X with several plugins open. The screen is cluttered with the plugins, and Logic Pro X is nearly inaccessible due to the many plugins, forcing the user to either close or move plugins.

THE MULTI-DEVICE PROTOTYPE

The system developed to explore the case is a context-aware multi-device system, for editing and producing music. The ecosystem for producing music in this case consists of a computer, a physical control surface, and an iPad as seen on Figure 5. The computer is running Logic Pro X, for producing the music using for example plugins to alter the sound of audio channels.

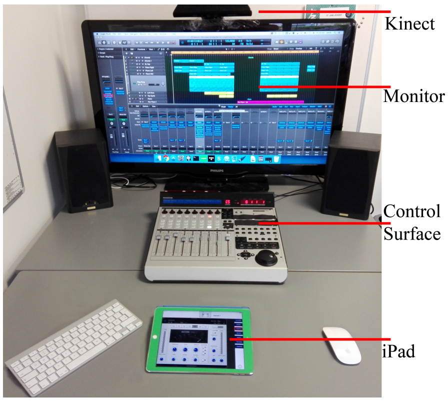


Figure 5: An illustration of the system setup, with a monitor at the back, speakers on each side, a physical control surface in front of the monitor, mouse, keyboard and the iPad. The iPad is tracked by the Kinect, and can be moved left or right to change channel on the physical control surface

The physical control surface is located in front of the computer, enabling fast controls and overview of the different audio channels. To reduce the complexity of the program view displayed on the screen, an iPad is used to display the plugins. The iPad can show one plugin at a time, enabling the possibility to view and interact with the plugin. The iPad is tracked by the Kinect, and by moving the iPad left or right, the system will change channel accordingly in Logic Pro X.

An example of the setup can be seen in Figure 5 and the system on the iPad can be seen in Figure 6. A video demonstration of the system can be seen at `https://goo.gl/GUGgtm`.

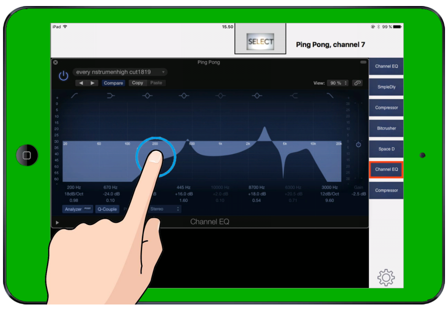


Figure 6: A picture of the iPad's screen, on channel 7 with the plugin *Channel Equalizer* open. The top bar shows the position of the channel by displaying the channel name and channel number. Furthermore, a "select" button is shown directly under the select "button" on the physical control surface. The right bar shows the available plugins for the selected channel, which the user can press to show the plugin on the iPad. The active plugin, on the figure the Channel Equalizer, is marked with a red border on the right panel.

Spatial Tracking

The iPad's position in regard to the physical control surface is tracked with a Kinect. By moving the tablet to the left or right, it is possible to change the selected channel, choosing between the eight channels displayed on the physical control surface, illustrated on Figure 7.

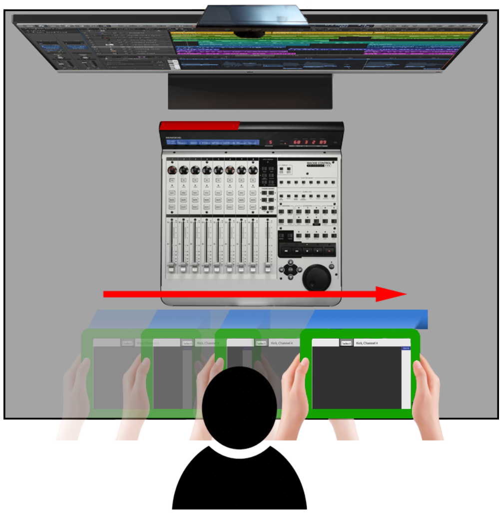


Figure 7: An illustration of the interaction of selecting channels in Logic Pro X by moving the iPad.

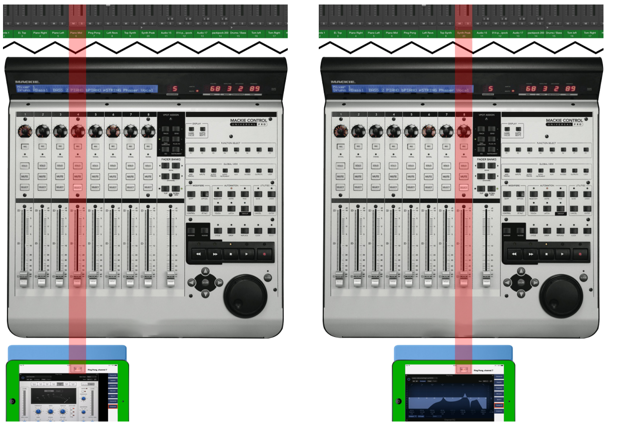
Moving the tablet left will select a lower channel, moving it right, a higher channel. When the channel is changed by iPad movement, it is illustrated by the channel number and name in the top of the iPad, as seen in Figure 6, as well as highlighting the corresponding "select" button on the physical control surface, and highlighting of the channel in Logic Pro X as seen in Figure 5. The channel can also be changed through Logic Pro X or the physical control surface. If the channel is changed in Logic Pro X or physical control surface, the iPad will display this selected channel, thereby overriding the spatial tracking. The spatially tracking is activated again when the iPad is moved, cancelling the override.

Figure 8 illustrates the tracked area in front of the physical control surface. The tracked area is divided into 8 subareas, corresponding to the channels on the physical control surface. When the iPad enters the tracked area, the channel will automatically change according to the location of the iPad.

The tracking provides another interaction method, making it possible for the user to work how they prefer, as well as potentially making it more effective or give a better overall music production experience.

Plugins on the iPad

When a channel is selected, the user has access to plugins added to this channel on the right side of the iPad. By pressing one of the plugin buttons, the corresponding plugin will be opened on the iPad. The opened plugin is now remembered when this channel is active on the iPad. Selecting another channel, either through Logic Pro X, the physical control



(a) (b)

Figure 8: An overview of how the tracking of the iPad works. The red channel represents the currently active channel. a) Shows the iPad being tracked to channel 4. This selection is reflected both on the iPad, the physical control surface and in Logic Pro X. b) Shows the iPad being moved to the location of channel 8, where this channel is now active on all three devices.

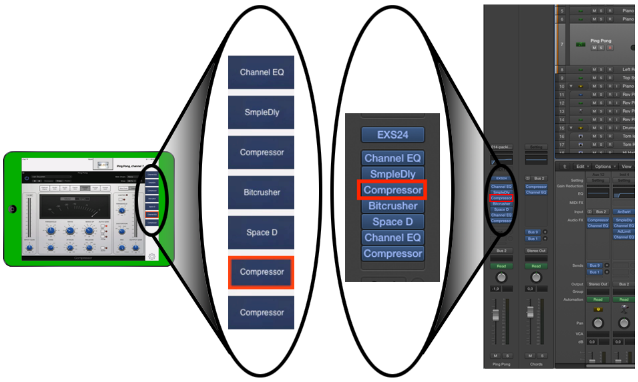


Figure 9: The figure illustrates the plugin list in both Logic Pro X and the system on the iPad. On the left, the corresponding plugin list on the iPad is shown, where the active plugin, the compressor, is marked with a red border. On the right, a section from Logic Pro X is shown, where the plugins are listed. The active plugin is marked by the system with a similar red border. The color of the buttons on the iPad are set to match the colors of the buttons in Logic Pro X.

surface or by moving the iPad, will hide the plugin. The plugins of the newly selected channel will then open. If no plugin has been opened on this channel, the iPad screen will be empty, only showing the top bar with information of the channel, and the right side bar with the list of plugins on the channel. Tapping another plugin button on the iPad will results in the old plugin of the channel being replaced with the new plugin.

Only one plugin can be opened at a time on the iPad, and only one plugin can be opened on the iPad on each channel resulting in the plugins being linked to the given channel. In this way, the iPad remembers which plugins are active on each channel.

As an example, the Channel EQ on channel 7 illustrated in Figure 6 is open on the iPad. If the channel changes away from channel 7, the plugin will close. If a plugin had been opened on the newly selected channel, this plugin will open. Otherwise the iPad will not display any plugins. When channel 7 is selected again, the Channel EQ will again open. This makes it possible for the user to continue their work with the plugin. In this way it is possible to work on plugins on different channels, and quickly change between them, to perform adjustments, by moving the iPad.

To aid the user keeping track of the active plugin, and the connection of this plugin on the iPad to the computer, the button for the active plugin is highlighted by setting a red border around the button, as seen on Figure 9. This is linked to the computer, where a similar red bordered box is set around the active plugin, as seen on the figure. This will aid the user identify the active plugin, and create a visual link between the iPad and the computer.

Visual Consistency

When creating an application as an extension to an existing program, the design needs to match that of the existing system to create a coherent user experience. In order to design for a coherent experience, Gestalt theory can be referred to, because the laws of perception from Gestalt theory can aid the design of systems that helps the users learn and interpret the system [9]. The law of similarity from the Gestalt theory has been used by matching the colors of elements of Logic Pro X and the physical control surface. The plugin buttons on the iPad was designed to match the colors of the plugin buttons in Logic Pro X as seen in Figure 9, in order to create a link between the functionality of these button. The coloring of the application frame was designed to match they gray color of the physical control surface, as seen on Figure 5 and Figure 6.

To create a visual link between the physical control surface and the extension on the iPad, the channel "select" button on the physical control surface, which lights up if the channel is selected, was drawn on the extension interface, as seen in Figure 8. Next to the "select" button on the iPad, the channel number and title is displayed. This virtual "select" button with text moves according to the movements of the iPad, such that the virtual "select" button will always be directly under the active "select" button on the physical control surface, to facilitate the linking between the two devices. This fits both the Gestalt law of Continuation and the Gestalt law of closure. The physical control surface can be perceived as extending down onto the iPad, however since there is a physical gap between the physical control surface and the iPad, our minds will close this gap implicitly [1].

Relation to the 4C Framework

From the perspective of the 4C Framework presented by Sørensen et al. in [10], our design falls within the Complementarity theme of interaction design, with a single user using multiple devices simultaneously. Within this theme, the system is an Extension, because the existing system is partly extended to the iPad. Complementarity was chosen, because it is interesting to study the effects of adding a radically different device, where the interaction and functionality is designed with the existing ecosystem in mind.

The multi-device system for this case consists therefore of the iPad, the physical control surface and the computer with mouse and keyboard.

Technical Implementation

The technical implementation of the system will be described here. The setup can be seen in Figure 5.

Screen Extension

The screen of the computer is extended to the iPad through Wi-Fi as a secondary screen, using the application Spashtop WiFi Display. Plugins from Logic Pro X are moved to the extended part of the screen, which is viewed and controlled on the iPad. The program handling the plugin movements is written in Swift 2.2. The screen extensions functions as a multi-monitor setup, where program windows can be dragged freely between the two monitors.

Communication with Logic Pro X and the Control Surface

To detect channel or bank changes made through Logic Pro X or the physical control surface, MIDI messages between Logic Pro and the physical control surface are intercepted and processed. The MIDI messages are intercepted by a separate program, which sends notifications to the main program when certain events arise. To be able to display information from Logic Pro X on the iPad, AppleScripts are used to read values and information directly from the interface of Logic Pro X. This information include which plugins are placed on the current active track.

Color Tracking

The iPad's position is tracked using a Kinect. The Kinect tracks the iPad using color tracking, as the iPad is tracked by a blue Apple Smart Cover. The iPad has also been colored green using a sticker around the screen, which is used if the Smart Cover is not recognized due to for example the lightning in the room or the angle of the cover towards the camera. The physical control surface is also tracked by the Kinect with color tracking using a red piece of paper which is mounted on the physical control surface.

The tracking is written in C++ and implemented with the libraries Freenect and OpenCV. Freenect is used for communication with the Kinect and gets a picture from the Kinect's camera, which can then be processed. The image processing is done using OpenCV. After converting the image to HSV values, the OpenCV method *inRange* is used to get a thresholded image, where pixels matching the given color range are colored white, others are colored black. The OpenCV methods *rode* and *dialiate* are used to remove the worst clutter and disorder. Then the method *findContours* are used to get a vector of a vector of points (a matrix of points), containing the contours of clusters of white pixels in the thresholded image. The largest cluster is then used, as position of the tracked object. The iPad's position is tracked by the middle of the calculated contour. With this approach, the Kinect image is first thresholded with the blue color for the iPad. If it is not found on the image, it is thresholded with the green color for the iPad border. Then it is thresholded for the red color for the physical control surface. The position of the physical control surface and iPad on the image is therefore known.

Spatial Channel Detection

To determine which channel the iPad should activate according to its location, the area in front of the physical control surface is divided into 8 subareas, representing the 8 channels on the physical control surface. When the iPad is moved to another subarea, the channel is changed accordingly. This is illustrated on Figure 8. To stabilize the channel shifting, preventing the channel to shift rapidly forth and back if the iPad is just at the border between two subareas, a margin for each subarea has been applied. Within the margin, the channel will not be shifted. The channel is therefore first changed, when then iPad is significantly inside the other subarea. The position of the iPad in relation to the physical control surface is also used to position the select button in the top of the iPad.

STUDY OF USE

In order to investigate the impact and use of the prototype, we conducted a study of use with five participants. The tasks and equipment was acquired with help from the music department at Aalborg University.

Participants

The study was conducted by four participants, two second semester students and two fourth semester students, from the music department at Aalborg University with the field of study "Popular Music and Sound Production". The participant are hereafter named participant 1-4 (P1-P4). They were all male, and between the age of 21 and 28. They all had extensive experience through their education with mixing tracks using Logic Pro X. All of participants had experience using touch screen interfaces, but two of them did not own a smartphone, however two of them owned a tablet.

Procedure

Each participant was asked to mix a multitrack Logic Pro X project, towards the quality of another song. This means that the first song should should be mixed to match the sound qualities and sound elements of the second song. This assignment was designed with the help from Mads Walther-Hansen from the music department of Aalborg University. The participants was given a Logic Pro X project, which did not have any plugins, and was just a raw project, with only the music tracks added. They were furthermore equipped with a sound file containing the song which they should mix the Logic Pro X project towards.

First, the participants got 25 minutes to mix a track with the usual equipment; the computer and physical control surface. This was to observe how they normally worked, and to focus the participants. After a short break, the participants got a short introduction and demonstration of our system. Hereafter, they got 40 minutes twice, with a small break between, to mix two multitrack Logic Pro X projects towards the quality of another song using our system. Afterwards, a semi-structured interview was conducted. Throughout the evaluation, a test facilitator was present to introduce the participant to the task, give the short introduction of the system before second part of the evaluation, and to reset the system if any bugs occurred.

Technical Setup
The system consists of an Apple iPad Air 2, an Apple Mac Pro, a Mackie Hui Universal Pro Digital Audio Workstation and a Microsoft Kinect for Xbox 360. The Mac Pro was equipped with a 3.5 GHz 6-Core Intel Xeon E5, 64 GB RAM and an AMD FirePro D700 6144 MB graphics card. The monitor used was a 42" Phillips Full-HD TV. The user was placed approximately 132cm from the screen. The physical control surface was placed between the user and the TV, approximately 42cm from the user. On each side of the TV, two Dynamic Acoustics BM5P speakers were placed which where connected to the computer through a Tehcnics SE-A800S Stereo Power Amplifier. The setup during the study can be seen in Figure 10.

Figure 10: Photo of the evaluation, where a participant is using the physical control surface and the iPad to interact with Logic Pro X on the computer.

Data Collection
Both the screen of the computer and the iPad in the study was recorded during the evaluations, and the users' interactions and speech were recorded with a Panasonic HDC-Z10000 camera, pointing down at the iPad, physical control surface and computer screen. A test observer was present during the whole evaluation to write down observations. The test observer wrote a log during the part were the participants used the system, and wrote a transcript during the interview. Afterwards the video material and screen recordings were reviewed, in order to refine, fill the holes, and expand the logs and reviews written during the evaluations.

RESULTS
Through the observations and recordings from the study of use, several interesting results emerged. The results have been grouped into themes in the following.

Multi-Device Use
From the study, it was clear that the users used the devices differently, in different pace and for different purposes. An example is P3, who used the physical control surface intensively, while P1, P2 and P4 nearly did not use the physical control surface. All participants used the computer screen as working area in the first part of the evaluation, where they did not have the iPad available. P1, P2 and P4's computer screens were all cluttered with plugins in the first part of the evaluation to a degree, where they had to move plugins to access Logic Pro X underneath. Several of the participants states, that they have earlier worked with multiple monitors to reduce this clutter, by e.g. having the Logic Pro X main window on one screen and plugins on another.

In the second and third part of the evaluation, every participant changed working style with the computer screen. P1 says about the computer screen: "It become more like a device for overview", indicating that the role of the computer screen has changed with the introduction of the new device. P3 states that he used the computer screen for the overview of the track, while using the plugins on the iPad. P2 says he looked away from the computer screen, down to what he was doing on the iPad, removing all other information from his gaze than the information needed for the current job. P2 also mentioned, that in normal use, the computer screen would often be filled with multiple plugins overlapping, resulting in problems with accessing the controls underneath, which is confirmed by the observations from part one in the evaluation. These statements combined clearly indicate that the use of the computer screen changed with the extension with the iPad, as the tasks were divided for the different devices. P1 did e.g. state that "I think it is cool that you can distribute the work tasks". They used the computer screen for overview, while the iPad was more of a working station. P2 described it as a funnel, where the computer at the top of the funnel had the full overview of all channels etc., the physical control surface in the middle of the funnel had a more refined overview of 8 channels, and the iPad in the bottom, which displayed information about one channel.

The use of the different devices was different for each participant, but some of the devices were used to interact with each other. To fully describe the multi-device system, there are 5 devices: The computer, the physical control surface, the iPad, the keyboard and the mouse. The iPad and physical control surface is extensions of the computer, while the mouse and keyboard are remote controls, in regard to the 4C framework [10]. Three of the participants (P1, P3 and P4) used either the keyboard or mouse as remote controls for the iPad, even though this behavior was not intended by us. P1 used the mouse to fine-tune a plugin located on the iPad, where a touch interaction with the finger did not provide the needed precision. P3 had to remove an element of a plugin, and therefore selected the element by touch and deleted using a keyboard button. P4 pressed a placeholder for a value, and entered the value on the keyboard. This could indicate that they view the newly added devices as so tightly coupled to the other devices with similar characteristics, that they use the same input devices in some cases. No participant did though notice the relation of the select button to the physical control surface.

P1 mentioned that the added number of devices could be utilized when mixing as a group of people, as the tasks could be divided to several devices, which could be used by different people. A system like this would then also be using the theme Collaboration from the 4C framework.

The users did use the devices differently. As mentioned, only P3 used the physical control surface, and their opinion of the system were also different. P3 thought that the touch interaction was especially good for the channel equalizer plugin, while P1 found it very less suitable for the channel equalizer. It can therefore not be a goal to extend a system, which every person using the system finds suitable, as each person has their own working style, methods and habits.

Spatial Tracking

The spatial tracking of the iPad caused problems, because the iPad was tracked using color. This color tracking was unstable when participants put their hands in front of the iPad, which happened when they interacted with the physical control surface, which was located between the iPad and the monitor. Even though most of the participants did not use the physical control surface much, they still used it to change which channels the physical control surface represented, as it could only show 8 channels at a time. The tracking works by finding the middle point of a given area, in this case the largest blue object, and finding the middle of this object. When a participant put his hand in front of the blue object, the size and location of the largest blue area changes, therefore the middle of this also changes, and the tracking will cause the system to change channel. P2 called the tracking a speed bump, because it took longer to change channel by moving the iPad, and then select the plugin on the channel, compared to using a mouse, he said: "I need to think about my choices and movements". P4 even compared the functionality and use of the tracking an obstacle course.

It was observed that the participants only used the iPad to change channels when they needed to interact with the plugins, and not to change channels in general. The participants mental model of how the tracking worked also caused problems. When a participant changed channel either in Logic Pro X or with the physical control surface, the iPad would reflect this channel change regardless of its spatial position. However this would sometimes cause problems for the participants when they wanted to change channel with the iPad later. An example is in part 3 of the evaluation by P2, where the iPad was located around channel 4, but the spatial tracking was overridden so the selected channel was channel 2 due to other channel selection than through the iPad. The participant wanted to change to channel 3, and drags the iPad intuitively to the right, which causes the iPad to change to channel 5. This makes the participant drag the iPad forth and back to locate channel 2, indicating that they thought of the iPad tracking as a relative tracking, in regard to the given selected channel regardless of the possible overridden spatial tracking.

P3 suggested expanding the tracking to include the entire table surface, and being able to change between all channels, instead of just the 8 channels on the physical control surface, and P4 suggested being able to have indefinitely many channels on the iPad, however being able to change between them with scrolling instead of spatial tracking of the device.

Use of Plugins on iPad

Several of the participants stated, that the interaction on the iPad was more intuitive and creative. P2 and P3 stated it was a very intuitive interaction, while P4 stated "... as production tool, it would be fine to have this intuitive approach to form sound images here [The iPad], and have the locked metaphors up here [Monitor]", also describing that it is intuitive that when he wants something to happen, he has to to this with the hand intuitively. Several participants emphasize the advantages of having the plugins in their hands. P1 says "I think it is a very cool thing to have it in my hands". He later states "Having it in the hands help me getting it close to me". P2 stated that it in a way was more intimate, having it all in front of him. The participant described the way of working with the iPad as more creative and abstract. P3 describes the iPad use as "a piece of water you stick your fingers in and stir", while P4 mentioned that it "could be like shape something like when you do artwork in clay". P1 states, that it "Gives a greater feeling of intimacy and form for control". P2 describes it, as a more physical experience. P1 also described that it "helped visualizing it a lot clearer", having the plugins in his hands.

Even though the image on the iPad is just a screen extension, the participants use several of the known interaction techniques from regular iPad apps. Both P1, P2 and P3 tried to zoom on a plugin by pinching, as they had problems with precision of a plugin on the iPad, or due to small icons and text. P1 and P3 succeeded with the zoom, as it was incorporated in Splashtop, even though we had not though of it. P4 tried to turn a plugins knob button clockwise with a circular knob turning-like gesture, as you would have done with a physical knob, even though it is known from Logic Pro X that you turn knobs by dragging vertically upwards and downwards.

Having the plugins on the iPad at close range did both reveal advantages and disadvantages. All participants had problems with the precision of the iPad, when adjusting plugins. P1 had problems fine tuning a plugin, saying "This is really hard". P2 said the use of plugins on the iPad was used with general strokes, instead of considered choices. P3 tried to adjust a value bar, but the value bar went to the bottom, even though the participant did not move the finger very much. P4 explains, that the touch function was not precise enough for him to tune the plugin, when e.g. changing the Q-value on a Channel Equalizer plugin, as he felt he could either choose a very high or very low value, but could not fine tune in between. From this it can be seen, that the iPad had some problems when it is needed to fine tune the plugins, as the touch implementation is not precise enough. Some interactions where not possible to do by touch, like hovering the mouse as P2 discovered.

Several participants stated that the plugins should be optimized for touch, especially due to the precision problems and size of text etc. These optimized plugins could utilize the touch interaction to support the creative and abstract interaction, where you like P4 describes it could be like shaping with your hands. They stated, that some plugins would be better than other. E.g. the Channel EQ's top part, see Figure 6, which contains a graphical representation, could be optimized for use with touch, supported by the statements of both P3 and P4. Some plugins works better on the computer, according to e.g. P1 saying "Some things are just much better having at your hands, and other thins are better with the mouse".

Speed of System and Simultaneous Precise Interaction

From the interviews and observations of the participants, there were problems with the speed and precision of the synchronization of the system. This resulted in participants becoming impatient, which was mentioned directly by all participants. P2 did however notice that this made him consider his next actions more. There seemed to be a latency between the computer and the video feed on the iPad screen, causing P1 and P4 to use specific plugins such as the compressor and the channel equalizer on the computer screen. Both mentioned that this was because they needed to be able to follow the live visualization of the beat in the plugins to be able to figure out precisely which changes they needed to make. A delay in this visualization was therefore the reason they used these plugins on the computer screen.

When a change of a plugin on the iPad was initiated, a loading indicator was shown in the middle of the screen. This indicator did not always disappear, however all participants ignored this and started interacting with the plugin regardless of the loading indicator. This also indicates that the participants became impatient while waiting for the system to load, and simply tried using it. P4 mentioned in the post interview that the speed of the system made him work slower, and that the frequent re-calibration made him loose his line of thought. He stated in the post test interview: "It destroys workflow more than it creates workflow".

Multiple times during each evaluation, the system would become out of sync in terms of which channel the program thought the iPad was located on. For example the system would display a plugin on channel 4, even if the iPad was line up in front of channel 5. This required that the test facilitator had to interfere with the evaluation and make the iPad re-calibrate using a re-calibration function in the system. All participants mentioned this imprecision during the post interview.

DISCUSSION

From the listed results, the discussion gives higher knowledge and insight in the area of multi-device extension. The result of this discussion is multiple guidelines for multi-device extension.

Input Methods

From the observations, it was clear that the functionality of the iPad was very well suited for some plugins, and not suited at all for other. Several people had problems with the plugins, which was not designed and optimized for touch interaction. This resulted in either a change of workflow, the user making changes on the computer screen or frustration because of e.g. lacking precision. Some users solved this by using either mouse or keyboard in relation to the iPad. This indicates that the users found the touch interaction inadequate for the interaction with the plugin. This especially happened when the plugin had to be fine tuned. Some interactions were however not possible to do at all using only touch, such as deleting elements or hovering with the mouse over a drop down menu. Some of the plugins had small text and symbols, making it hard for the participants to properly read the plugin. Several participants stated in their interview that they would have preferred plugins, which were optimized for touch, as this would have improved their experience of using of the system.

Some of the participants tried to use some of the interaction previously known to the device, in this case the iPad. They tried e.g. to pinch to zoom in and out. The system used for the remote screen, Splashtop, handled this with success, making the users able to zoom in on the plugins which had small text and icons, making it possible to use the plugin even though it was not optimized for the iPad.

From this, we have deduced the guideline that content, input methods and interactions should be device-optimzed and support common device-specific interactions, meaning that the content must be readable, interactable and seem to fit both visually and interactively for the device. The input methods for the device should be device-specific and optimized. It should support interactions the user know from prior experience with similar devices, e.g. swipe, pinch and drag gestures for iPads. This is to ensure the user can perform the required interactions given by the content shown on the device. It should also be optimized in sizes etc. to give the user an experience of the content being meant for the given device, and ensure that no text or symbol is too small or large to be read. This contradicts some of the known theories of muilti-device development, which emphases the importance of consistency [8]. We argue, that different devices has different purposes, interactions, sizes etc., and the content should therefore be optimized for the specific device, as long as they are internally consistent on the device. The content could therefore look different on different devices, as well as the interaction should differ. To aid the user in linking the elements on the different devices, the elements could have similar appearance, as e.g. colors of buttons and shape, as is the case with the plugin buttons, where the colors, shapes, sizes and interactions fits the common design of the device.

Different Devices for Different Purposes

From our results, it is clear that introducing another device in the ecosystem of music production changes the working style and habits of experienced users. It was clear that as soon as the iPad was introduced, the computer screen became an overview device for all of the participants, where the user could manage the production, while performing changes to each individual track using the iPad. However the functionality of the introduced device does not necessarily fit all tasks within a multi-device context. This was due to the design of the plugins. Similarly, there were other parts of the music producing workflow, which was not suitable to be done on an iPad, such as fine-tuning audio and panning levels, which participants did using mostly mouse, but also the physical control surface.

The different devices provides different functionality and advantages. The computer screen makes it possible to keep the overview, where the iPad would not give the same overview due to the size of the screen. The physical control surface provides easy adjustment of e.g. volume and panning, but has a very limited display, and a fixed set of input button, which does not make it very suitable for either keeping the overview of a large production or make changes to a lot of different plugins.

This division between tasks and device purposes therefore yields the guideline, that the purpose of the device must fit the device's functionality, meaning that different devices has different advantages and disadvantages, and the purpose of each device in the ecosystem should therefore be thought trough, and the tasks distributed among the different devices.

Relative Tracking of Device

The iPad was spatially tracked in relation to the physical control surface, as the area in front of the physical control surface was divided into eight spaces each with a corresponding channel. However it was clear that the participants thought the iPad was tracked in relation to the selected channel in the system, even though the tracking had been overridden by pressing on e.g. the physical control surface. It appeared that the participants felt it more intuitive to move the iPad left or right in order to change channel up or down, not according to the physical control surface, but the selected channel. It was however observed that participants had problems hitting the desired channel when using the iPad to change channels, where it might have been easier for the participant to control the channel shift by left or right movements instead of needing to place the iPad at a specific area in front of the physical control surface.

Through the observations it was discovered, that much of this confusion was due to the users override of the channel selection, either in Logic Pro X or trough the physical control surface. This leads to the guideline, that overridden or changed behavior should be marked very clearly to the user, ensuring the user know that the given behavior is changed or overridden. This could possible have prevented many of the users tracking issues, possibly changing their opinion about the spatial tracked system.

Simultaneity

The users used the system simultaneously, moving the tablet, while looking on the screen, or adjusting a plugin while listening to the changes. The system had however problems with the delays, both in the systems reactions of user input, but also for the video feed for certain real-time dependent plugins. The delays were according to the participants a reason why, they would not use the system in their future work, as it halted them in their work and line of thought. The latency of the video feed made the system unsuitable for certain plugins, limiting the use of the system.

From this, a guideline can be deduced: The synchronization between the devices in the ecosystem should always be stable, fast and simultaneous. This is to ensure the coupling of the devices, give a clear indication of the synchronized devices, and allow the user to have the same feeling of speed and access to content on each devices. This is also ensuring that all devices have an equal feel in regard to one device not behind behind other devices in synchronization. The simultaneous synchronization would also ensure the possibility of working with real-time dependent content.

Robustness

From the study of use, it was clear that the participants were frustrated by the problems which occurred in the system. Participants said their workflow was broken by the errors which occurred in the system, both when the test facilitator had to either restart the program or resync the position of the iPad. Through the interview the errors, delays and tracking problems was repeatedly in focus by the participants, being cause to much of their critique of the system. Through the observations, it was clear that the errors halted the users in their interaction, causing confusion and frustration and a need for the test facilitator to step in. This also happened when the channel was changed due to participants reaching over the iPad, possibly limiting the use of multiple devices, as it happened often while reaching for the physical control surface.

From the above discussion, a guideline can be deduced: The system should be robust, avoiding errors and crashes in order to make it possible for the users to use the system without having the need to restart the program or deal with errors, to ensure a steady workflow without interruptions and frustration. This is of course a guideline which could have been deduced for all systems, but we find it especially important for multi-device extension systems, as one crash on one device can effect the workflow of many devices in use. The users also has to keep focus on multiple devices, which makes it important that the system is robust.

CONCLUSION

Through this research project a multi-device system was developed in order to gain insights about the principle of extending in the theme Complementarity in the 4C framework. The observations from the study of use helped gain valuable knowledge about which parts of a multi-device system are especially important, and must be kept in mind when designing for such. The system had problems with the tracking and synchronization, resulting in frustration for the participants, which could affect the interpretation of the system. Despite of the errors, the study gave insight in to both the case implementation, but also their use of multi-device extension in general. We believe that the results from this study of use can help guide the future work within this area of multi-device ecosystems.

It can be concluded that there are challenges when designing a multi-device extension system, which should be kept in mind when designing such systems. Some of these challenges are easier to deal with if it is possible to articulate them. Using the 4C framework, this is to some extent possible, however there are still many corners of multi-device systems, which are to be shed a light on. Through our research, we have developed guidelines which are our contribution to the field of multi-device systems. These along with the additional knowledge is this paper's main contribution to the field and should serve as guidelines, when developing multi-device systems with an emphasis on the extension principle.

The first guideline is that content, input methods and interactions should fit the specific device, and be designed with the opportunities and limitations of the device in mind. Moreover, as the second guideline states, the purpose of each devices differ, and the functionality of each device should be taken into account to best facilitate the desired role of each device in the ecosystem.

When designing for a digital ecosystem which involves spatial awareness, the third guideline states that if behavior is changed or overridden, this should be very clear for the users. Overridden behavior could, if not marked very clearly, cause unexpected behavior, which would make the system feel counter intuitive.

The fourth guideline is, that synchronization between devices in an ecosystem must be stable, fast and simultaneous in order for the all devices in the ecosystem to feel updated in relation to each other. This also affects the information which has to move across devices, which is less useful if it is delayed. In multi-device ecosystems it is especially important, that the system is robust across all devices which is the last guideline. If an ecosystem, or part of an ecosystem is fragile or unstable, it will cause the ecosystem to seem fragmented.

These guidelines can be used to extend the 4C framework, and should be kept in mind for further research and development of multi-device extension systems. However it should be noted that the guidelines are not necessarily generalizable. Furthermore, the seven other principles of the 4C framework should likewise be explored with the goal of providing guidelines for these, to strengthen the framework, which would then not only be a framework to categorize and describe multi-device systems, but also support researchers and developers with design guidelines for use in their systems. We see big potential in such an extension of the 4C framework, which could make the use of the framework widespread.

LIMITATIONS AND FUTURE WORK

Even though the developed system was functional, errors occurred, and limitations had to be taken into account. The two main reasons for limitations and error sources were the Kinect color tracking and the closed sourced, non public API nature of Logic Pro X.

As for the color tracking with the Kinect, the system was light sensitive causing tracking flicker. Most of the interaction with Logic Pro X has been implemented using AppleScripts, as Logic Pro X is a non-scriptable, closed source program, with no open API. This meant that the interaction with AppleScript did not go as smooth as hoped, as Logic Pro X was not designed for this interaction. AppleScript was slower than expected, resulting in a less smooth interaction.

These limitations and error sources have been handled as best as possible, but they still caused some frustration and errors for the users in the study. It is though still believed that the results from the study is reliable and usable, just as the underlying errors in the system is kept in mind. For future research, the guidelines deduced from this study should be evaluated with a new system developed with these guidelines in mind. This would give even greater knowledge of the principle extension. As the guidelines from this study are not necessarily generalizable, a different case for a study could also yield different results with different or additional relevant guidelines for the extension principle.

To fully utilize the 4C framework, guidelines like those provided in this paper, should be provided for the 7 other principles. This will give a great knowledge of multi-device systems, and aid both developers and researchers in their use and study of multi-device systems.

ACKNOWLEDGMENTS

We would like to thank our supervisor Jesper Kjeldskov for guidance and feedback throughout the project. We would also like to thank assistant professor Mads Walther-Hansen for sharing his knowledge of music production, mixing and further music-related feedback and advice in the project as well as providing equipment and helping with designing the assignments for the evaluation. At last we would like to thank the participants for helping evaluation the product.

REFERENCES

1. Dempsey Chang, Laurence Dooley, and Juhani E Tuovinen. 2002. Gestalt theory in visual screen design: a new look at an old subject. In *Proceedings of the Seventh world conference on computers in education conference on Computers in education: Australian topics-Volume 8*. Australian Computer Society, Inc., 5–12.
2. Xiang 'Anthony' Chen, Tovi Grossman, Daniel J Wigdor, and George Fitzmaurice. 2014. Duet: exploring joint interactions on a smart phone and a smart watch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 159–168.
3. Peter Hamilton and Daniel J. Wigdor. 2014. Conductor: Enabling and Understanding Cross-Device Interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM.
4. Steven Houben, Paolo Tell, and Jakob E Bardram. 2014. Activityspace: Managing device ecologies in an activity-centric configuration space. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*.
5. Charles Martin, Henry Gardner, and Ben Swift. 2014. Exploring percussive gesture on iPads with ensemble metatone. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 1025–1028.
6. Heidi Selmer Nielsen, Marius Pallisgaard Olsen, Mikael B. Skov, and Jesper Kjeldskov. 2014. JuxtaPinch: Exploring Multi-device Interaction in Collocated Photo Sharing. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, 183–192.
7. Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp: Spatially-aware mobile displays for ad-hoc around-the-table collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 45–54.
8. Claire Rowland, Elizabeth Goodman, Martin Charlier, Ann Light, and Alfred Lui. 2015. *Designing Connected Products: UX for the Consumer Internet of Things*. O'Reilly Media, Inc.
9. Karen Smith-Gratto and Mercedes M Fisher. 1999. Gestalt theory: a foundation for instructional screen design. *Journal of Educational Technology Systems* 27, 4 (1999), 361–371.
10. Henrik Sørensen, Dimitrios Raptis, Jesper Kjeldskov, and Mikael B. Skov. 2014. The 4C framework: principles of interaction in digital ecosystems. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 87–97.
11. Martin Spindler, Wolfgang Büschel, Charlotte Winkler, and Raimund Dachselt. 2014. Tangible displays for the masses: spatial interaction with handheld displays by using consumer depth cameras. *Personal and Ubiquitous Computing* (2014).
12. Martin Spindler and Raimund Dachselt. 2009. PaperLens: advanced magic lens interaction above the tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*.
13. Anna Xambó, Eva Hornecker, Paul Marshall, Sergi Jordà, Chris Dobbyn, and Robin Laney. 2013. Let's jam the reactable: Peer learning during musical improvisation with a tabletop tangible interface. *ACM Transactions on Computer-Human Interaction (TOCHI)* (2013).

Development of System 4

This chapter describes implementation details of the system, as well as the first version of the system. This version was developed further into what became the final version of the system, which was evaluated in a study of use.

4.1 First Version of Project

The first idea of using spatially tracked multidevice systems in music productions was, to track an iPad and its position in regard to the physical control surface with a Kinect. It would then be possible to assign a plugin to either the right or left side of the physical control surface. When the iPad was on the left side of the physical control surface, the assigned plugin would be shown. Likewise the plugin assigned to the right side would be shown when the iPad was on the right side of the physical control surface. The interaction can be seen illustrated in Figure 4.1. A plugin was assigned by dragging it down to either the bottom right or bottom left corner of the computer screen. It was hereafter either replacing the plugin shown on the iPad, or hidden until it should be shown, according the the iPad's location.

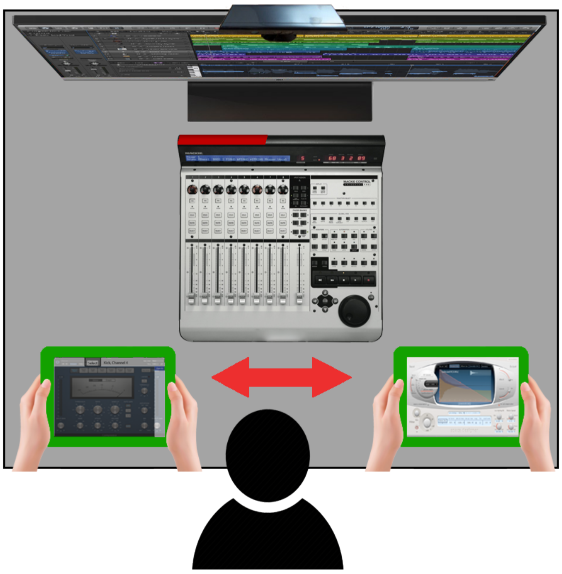


Figure 4.1: Illustration of the first version of the system. The iPad is tracked in relation to the physical control surface. On the right of the of the physical control surface, one plugin is showed, while another is shown on the other side.

A working version was developed, but through this process the idea of using tracking and plugins further, as used in the final prototype, emerged. The code base was therefore used as a starting point for the new direction of the project. Many of the complicated parts of the program, such as the tracking of the iPad and the movement of plugins movement from one screen to another, was preserved as these parts are largely the same in the current solution as in the former.

4.2 Final Version of Project

The final version of the project differs from the first version in that our program is closer integrated with the existing ecosystem. The iPad can display plugins from the channels in Logic Pro X, where in the first version, arbitrary plugins could be assigned to be shown when the iPad was either on the left or right side of the table. Furthermore it is possible to change between plugins directly on the iPad, where in the first version, the plugins took up the entire screen space on the iPad. An illustration of the final version can be seen on Figure 4.2. Here the iPad can show the plugins of each channel mapped to the physical control surface, depending on the iPads spatial location. If the iPad is located beneath the third channel on the physical control surface, the iPad will show the list of plugins added to that channel. A plugin from this list can be opened and interacted with on the iPad. If the iPad changes position to another channel, the system will close the current plugin, and remembers which plugins were active on the iPad on each channel. Thus if the iPad changes position to a channel where a plugin was open earlier, this plugin will reopen when the iPad enters the appropriate spatial location.

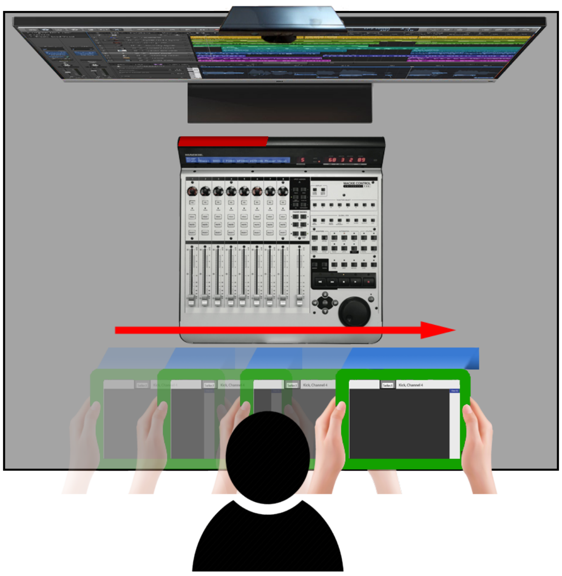


Figure 4.2: Illustration of final version of the our system. The iPad is tracked in relation to the physical control surface. According to where the iPad is located, the iPad will show one plugin, and the list of plugins on the selected channel.

A visual link between the devices has been created by adding a "select" button to the iPad. This button will move when the iPad is moved according to the physical control surface, so that the "select" button on the iPad always is beneath the appropriate "select" button on the physical control surface, which lights up when the channel is active. The selected channel is also marked in Logic Pro X with lighter background color than the unselected channels. This visual mapping is illustrated on Figure 4.3.

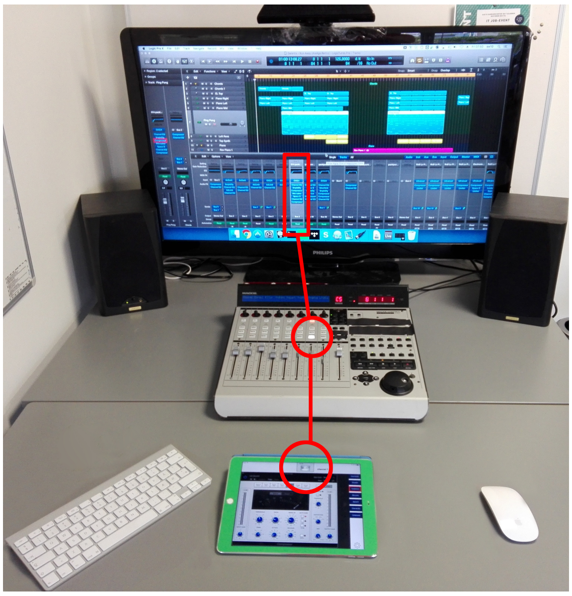


Figure 4.3: A photo of the full setup of the final version of the system. The mapping of the active channel has been marked for an easier overview.

4.3 Color Tracking with Kinect

The color tracking is implemented with a Kinect mounted on top of the monitor, pointing down at the table. The setup of the Kinect can be seen in Figure 4.4.

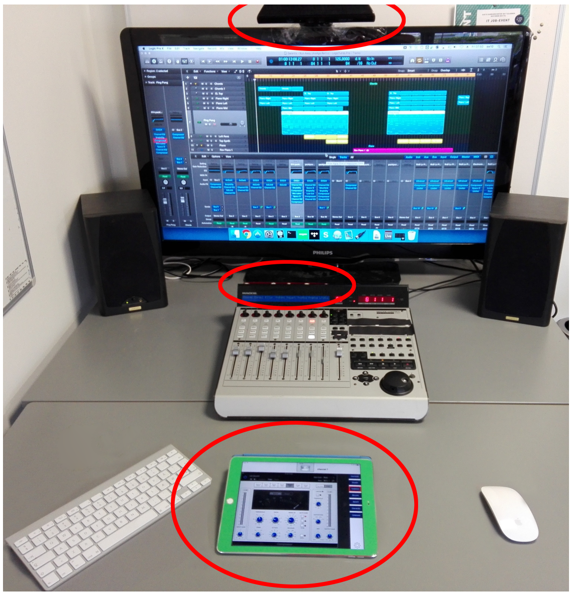


Figure 4.4: Setup of the system. The Kinect can be seen on top of the TV, pointing down at the table, marked with red. The red paper on the back of the physical control surface can be seen marked with red in the middle of the photo. The blue Smart Cover and the green sticker on the iPad can be seen marked in the bottom.

The area in front of the physical control surface is where the iPad is being tracked. This area is divided into 8 sub-areas, each representing a channel on the physical control surface. The currently selected channel has an additional margin, to the surrounding channels.

This is to reduce rapid channel changes back and forth, if the iPad is located on the edge between two channels. The iPad must not only exceed the border of the surrounding channel, but also exceed the margins. If the margins are exceeded, the channels are changed, and the margins are now set for the newly selected channel. An illustration of the tracked area divided into sub-areas can be seen on Figure 4.5.

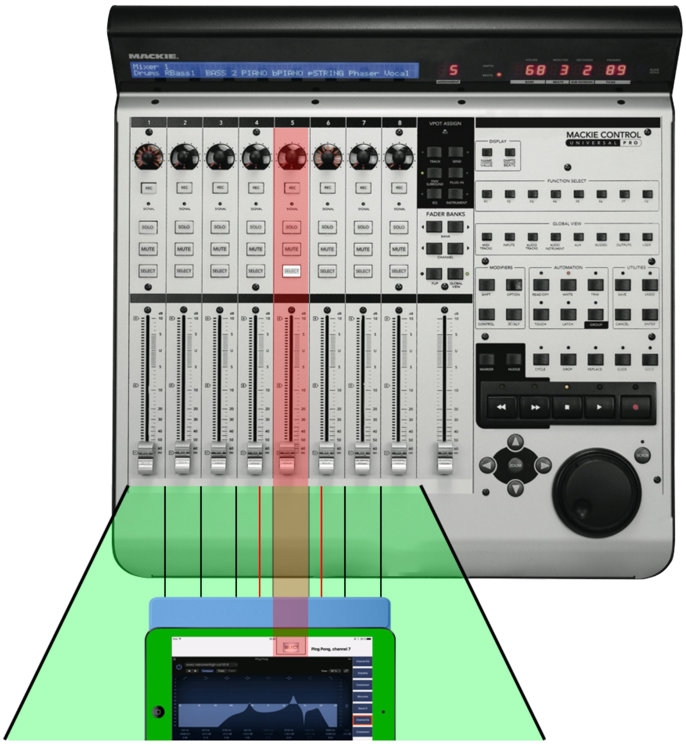


Figure 4.5: Overview of the color tracking, where the tracked area is marked with green, and the vertical black lines represent the sub-areas. Red lines indicate the margins, which the iPad has to exceed to change channel.

4.3.1 Communication Between OS X and the Kinect

The communication with the Kinect is implemented using the Freenect library. The image data, received from the Freenect library, was processed using OpenCV. As the Kinect is designed by Microsoft, establishing the communication between the Kinect and the mac computer was a non-trivial problem. Different versions of OS X did also result in different problems. Therefore we chose to focus the development on the currently newest version, OS X El Capitan. It was necessary both to install libfreenect and libusb to establish the communication. Hereafter OpenCV was built and installed using cmake. To complete the installation, different folders, e.g. lippicv, had to be copied to the local user library and include folder. Hereafter, the connection between the Mac and Kinect was established, even though it was more stable on some computers than others. This process was a combination of several online unofficial tutorials, which each solved individual problems, but no tutorial was found, which fully worked with both Freenect and OpenCV (even though stating so) on the newest OS X version, therefore the need of the combination of the tutorials.

4.3.2 Color Recognition

The color recognition is implemented with a C++ module. The picture from the Kinect is processed using OpenCV. The implementation recognizes the largest area of a color, to identify the location and dimensions of the object to track. To track a color, 6 values are given - two sets of HSV values. Tree lower bounds and three upper bounds of Hue, Saturation and Value values. The image from the Kinect is then thresholded using the OpenCV method *inRange*, which will return a thresholded image where all pixels of the original image lying outside the given HSV values were colored black, and all pixels within the HSV range was colored white, as seen in Figure 4.6. To reduce the worst clutter and disorder of the image, the OpenCV methods *rode* and *dialate* was used. This newly thresholded and cleaned image was then analyzed for groupings using the OpenCV method called *findContours* was used, which returned a set of contours of grouped white pixels, as seen in Figure 4.6. The largest elements of the sets of contours were found by the area of their surrounding rectangle. The object must be above a certain minimum area limit to be identified, to reduce false positive of very small areas, if the object to track is not present in the picture.

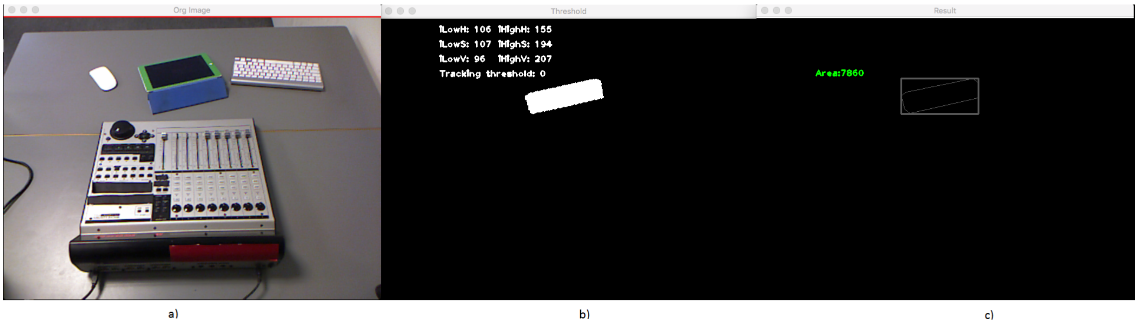


Figure 4.6: Illustrates how the color tracking works, by (a) having an original image of the table, as seen on the left. (b) A thresholded image can be seen in the middle, where all pixels within a specific color range (blue in this example) are colored white, all other are colored black. (c) from the thresholded image, the contours of clustered white image are found. The biggest area of the rectangle around the cluster is marked as the tracked object.

4.3.3 Color Tracking Module

The color tracking is implemented as a C++ module, which tracks three colors: The red color of the physical control surface, the blue color from the Apple Smart Cover and the green color from the iPad sticker. The tracked object can be seen in Figure 4.4. These colors are repeatedly tracked in a thread, tracking the colors every 100 ms. For every 100 ms a new image is captured using the Kinect. The image is analyzed using the above stated method, to receive the position of the physical control surface and the blue Smart Cover. If the cover is not found, the iPad is instead tracked using the green sticker. If both the physical control surface and the iPad are found, the position of the iPad is found in relation to the physical control surface, to find the channel area the iPad is located in. If the iPad has exceeded the margins for channel shift, the new result is stored. The main program can then read the result when needed. As the main program is written in Swift, is cannot directly call the C++ module. Using a bridging header, it was possible to write a wrapper in Objective C++. This wrapper can be called by Swift using the bridging header, and the wrapper can call the C++ code, and by this return the result from the tracking module to Swift.

4.4 Plugins on the iPad

The goal was, to show a specific plugin on the iPad instead of the computer screen. It should be possible to interact with the plugin, to change the values etc. To facilitate this, a lot of screen and window sharing applications was explored, to get the knowledge of the available functionality. The interaction of plugins in Logic Pro X is mainly based on mouse press and mouse dragging. A lot of applications were therefore discarded, as they during drag on the iPad did not drag the computer mouse, but just moved the mouse without holding the left mouse button down. Other screen sharing applications were to slow to either respond or show the video feed, and was therefore also discarded. Some applications would only show the whole computer screen, and not either an virtual secondary screen, or a specific window, which did not fit the purpose of move the plugin away from the screen.

The benefits and drawbacks of building a screen sharing software specifically to this project was also investigated, possibly with OpenVNC as starting point. At the end, one application, Splashtop WiFi Display, was found acceptable, making it possible focus on other parts of the development instead of developing a screen sharing software. Splashtop works by extending the computer screen with a secondary screen. This screen is then, by a WiFi connection, transmitted to the iPad. When dragging on the iPad, the mouse will drag on the computer, facilitating the interaction with Logic Pro X plugins. When displaying a plugin on the iPad, the window is dragged from the computer screen to the side of the screen, over to the secondary screen. To use the full area of the screen, each plugin is resized before being moved to the secondary screen. Usually AppleScript or other similar tool could be used to programatically resize and move windows in one step. However this was not possible, as Logic Pro's own window handling prevented this. The windows were therefore resized by implicitly defined C functions. It was not possible to move them by these implicitly defined C functions, so the plugin windows are moved by programatically moving the mouse, pressing and holding the top of the window, and moving the window to the secondary screen.

This is no optimal solution, but without further access to the code of Logic Pro X, this was found the best solution. When a plugin needed to be hidden, because no plugin was attached to the active channel, the plugin window was moved all the way to the bottom of the secondary screen, out of sight. It is then dragged up again when needed. If a new plugin should be shown, the fastest way to change the plugin was to activate the build-in Logic Pro feature to link a plugin, meaning that the linked plugin window will be replaced with a new plugin if pressed on in the program. The system would therefore activate linking, press the correct plugin button, and unlink the plugin window again.

4.5 Implementation Details

This section describe some selected details in the implementation, which in combination help the system have desired functionality. None of these details were in it self essential for the program, and is small features which combined gives value to the system.

4.5.1 Rouge Windows

Rouge windows are Logic Pro X windows which are moved to the second screen by other means than through our program. An example is if the user moves a plugin window to the second screen using the mouse. The program will automatically detect this, and move the window back onto the computer screen. This can also happen if the program needs to be restarted, where it will detect the currently active plugin on the tablet as rouge, and move this back to the computer screen. The reason for this behavior is that the the functionality regarding moving windows are based on movements and clicks with the mouse, which will not work if the Logic Pro X windows are covered with other program windows.

4.5.2 Marking of Currently Active Plugin

When the user presses a plugin button on the tablet, the button is marked with a red border, to indicate that it has been pressed, and which plugin is active on the screen. The same plugin will be marked on the computer screen in the list of plugins of the active channel. The program remembers which plugins are active on each channel, so if the tablet is moved to change channel, the marking of will change to the correct active plugin on the tablet. When a plugin button is pressed, the red border appears immediately to give the user visual feedback. A spinning icon is also added to the middle screen, to indicate that the program is working.

When the user changes channel, the list of plugins has to be updated, and also here a spinning icon is added to indicate the loading state.

4.5.3 Mouse Governor

Because much of the window movements happen by moving the mouse programmatically, a mouse governor has been implemented. The program keeps a buffer of mouse positions, and when needed, it can move the mouse back to a certain position. An example of use is, if a plugin window is moved to the second screen, it would be counter intuitive for the user if the mouse was located in a different position, because he pressed a button on the tablet. Therefore the mouse is moved back to its original position when the window has been moved.

4.5.4 Second Screen Background Color

An AppleScript is used to set the background color of the tablet to black. This is because the border of all Logic Pro X plugins are black, and therefore they seem embedded in the program window on the tablet. If more colorful backgrounds where used, it would be more clear that the tablet is simply an extension of the desktop on the computer. Using the black color, an illusion is created, where it seems that the program is actually running on the tablet, while in reality it is running exclusively on the computer, and is only shown on the tablet.

4.5.5 Position of Select Button on Second Screen

To add to the feeling of the program being an extension to the current ecosystem, a select button has been added in the top of the tablet screen. This select button looks identical to the select buttons on the physical control surface, and moves according to the tablet position. If the tablet moves right, the select button moves left and vice versa. The idea is to indicate the connection between the physical control surface by displaying the select button underneath the current active channel.

4.5.6 Integrating Logic Pro X in the System Through AppleScripts

To be able to get information from Logic Pro X such as which plugins are added to each channel, and which channel is currently active, an API is needed. Apple has however not provided an API for Logic Pro X. Some OS X applications are scriptable, meaning they respond to certain AppleScript commands, and thus can be interacted with easily through these. Logic Pro X was however not scriptable either. Therefore the only solution to obtain this information was to use AppleScripts to read values directly from the interface of Logic Pro X. An example of such an AppleScript can be seen in Listing 4.1. This AppleScript presses a certain button in the plugin window currently active on the tablet. First the script loops through each active process on the computer and finds Logic Pro X. Then it loops through each window of Logic Pro X, and finds the window which is located on the tablet by comparing x and y values. The button to be pressed is always element number 3 of a plugin window, which is set next. At last the button is pressed.

```
set xMin to 0
set yMax to 200
set goal to 0

tell application "System Events"
  repeat with theProcess in processes
    if not background only of theProcess then
      if name of theProcess is "Logic Pro X" then
        tell theProcess
          set theWindows to windows
          repeat with theWindow in theWindows
            set windowPos to the value of attribute "AXPosition" of theWindow
            if item 1 of windowPos > xMin and item 2 of windowPos < yMax then
              set linkBtn to item 3 of UI elements of theWindow
              set presentVal to the value of attribute "AXValue" of linkBtn
              if goal is not equal presentVal then
                click linkBtn
              end if
              exit repeat
            end if
          end repeat
          exit repeat
        end tell
      end if
    end if
  end repeat
end tell
```

Listing 4.1: An AppleScript, which presses a certain button in a certain program window.

This is a rather unstable, slow and complicated procedure, as AppleScripts in general are not reliable, because programs being read by AppleScripts can return wrong values under certain circumstances. To mitigate this, if an AppleScript returns an unexpected value, the script is simply called again. If the second call also gives an unexpected return value, the program will continue without it, in order to not block the entire system.

The need of AppleScripts does also slow down the system, as AppleScript does not evaluate as fast as a compiled program using an API, due to the need of tree traversing the interface.

Future Work 5

The developed system had a lot of both advantages and drawbacks, which was revealed in the study of use. Furthermore, the exploration of the 4C framework gave valuable insight, but did also reveal areas which should be studied more. Therefore, this section is divided into future work for the system and future work for the research of the 4C framework.

5.1 Future Work of System

If the developed system should be taken further, there are some critical issues, which should be resolved. The system should be more stable and robust, eliminating the errors which now occurs in the program. This includes the synchronization problems where the iPad went out of sync, the errors which crashed the program, the errors where spinning icons did not disappear and the errors where plugins were not moved correctly. Furthermore, both the speed of the window movement and the speed of the video feed should be improved further. As of functionality, the system should be further developed in regard to the interaction of the plugins. The plugins should be optimized for touch, both in sizes and interaction, as the participants had problems with the precision of their adjustments of the plugins. Some participants suggested a more abstract touch interaction, where they could shape the sound image with their hand. Such an interaction could be interesting to implement, and observe how the users would use such a system.

5.2 Future Work of Research

The extension of the 4C framework with the guidelines is a powerful addition to describe some of the challenges for multi-device extension systems. To support this addition, we suggest similar guidelines are made for the seven other principles of the 4C frameworks. Several of these guidelines could overlap, giving new results of, which principles overlaps and how. If all principles had guidelines, the 4C framework could fully help the developer with not only describing and informing multi-device system, but also aid the developer in the design and development of a successful multi-device system by providing the guidelines. The guidelines provided in this study are not final - an other implementation of an other system would properly have revealed other guidelines. Therefore it could give interesting results to develop a system with these guidelines as point of origin, and compare these results to the results of this study, verifying and refining the guidelines.

Conclusion 6

Personal digital ecosystems are continuously expanding, including an increasing number of devices. The devices are used both sequentially and simultaneously. To aid the development and research of such multi-device ecosystems we explored the principle of multi-device extension with the case of music production.

Out aim was to answer the two research questions:

> RQ1: "*How can a cross-device system be designed, which facilitate interaction during music production?* "

> RQ2: "*How can a spatially tracked multi-device system be developed to assist music production?* "

To answer the research questions, we have implemented and evaluated a system for extending the music production ecosystem with a spatially aware tablet. This tablet complement the existing devices (in the study a physical control surface and a computer with mouse and keyboard), adding new interactions and functionality. A study of use was conducted, to observe how expert users used the system, and get their opinion of the system and ideas.

The tablet enabled the user to change channel in their music production program, Logic Pro X, by using the tablets spatial location in relation to the channels on the physical control surface. On the tablet, information about the current selected channel was shown, including the channel number, name and the list of plugins on the channel. By pressing a plugin button on the tablet, the plugin will open on the tablet, enabling the user to adjust and interact with the plugin on the tablet instead of on the computer screen. To aid the users in understanding the link between the tablet and the other devices in the system, the buttons are made to look like those in Logic Pro X. When a button is active, a red border marks the active plugin button on both the tablet as well as in Logic Pro X. A "select" button from the physical control surface are replicated on the tablet, and located just below the actual button on the physical control surface at all times.

The result from the study was five guidelines for multi-device extension systems, extending the 4C framework presented by Sørensen et al. [Sørensen et al., 2014]. The first guideline is, that the content, input methods and interaction should be device-optimized, as different devices has different functionality, advantages and disadvantages. A touch interface should e.g. not require a mouse hover or right click, as well as the contents size and shapes should be optimized for the given device. The second guideline is, that the purpose of the device should fit the device's functionality. A device with a small screen should not have the purpose of an overview screen, and the devices should therefore in a multi-device extension system have different roles according to their functionality. The third guideline is, that overridden or changed behavior should be marked very clearly, leaving no doubt for the user, that the behavior has been changed or overridden. This was due to several of the participants in this study had problems when the spatial tracking was overridden by pressing on the physical control surface or in Logic Pro X, making them select wrong channels when re-activating the spatial tracking through tablet movement. The fourth guideline is, that synchronization between the devices in the ecosystem should always be stable, fast end simultaneous. This is to ensure, the user has a fluent experience of the content across of the devices, as well as having the same feeling of speed and access to content on all devices. The last guideline is, that the system should be robust and avoid errors and crashes. This is of course a goal for all systems, but is was found especially important for multi-device extension system to ensure a fluent flow for the user, as well as a feeling for the use that the devices are connected.

With this system in mind we conclude that we have answered the two research questions by showing that a system for music productions could be implemented. We have researched how a spatially tracked tablet can aid music producers in their work. The five guideline aid further research and development in the area, as they provide important knowledge of certain elements of multi-device extension systems. The guidelines are not proven to be generalizable beyond this system and case, but further research and development can use them as inspiration, and a later study could try to generalize the guidelines with other case studies. The guidelines can be seen as an extension to the 4C framework, as the framework helps to describe and categorize multi-device systems. With the guidelines, the framework could also aid the design of such system.

We think this area of research has great potential for further research, as multi-device ecosystems are of great importance and are in continuous development. The developed system shows how multi-device spatial tracked extension systems can be implemented and utilized in a music production, and we see a large number of other use-cases, techniques and types of multi-device systems to be explored, as well as further extension and utilization of the 4C framework.

Bibliography

Aumi et al., 2013. Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson and Shwetak Patel. *Doplink: Using the doppler effect for multi-device interaction.* In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, p. 583–586. ACM, 2013.

Czerwinski et al., 2003. Mary Czerwinski, Greg Smith, Tim Regan, Brian Meyers, George Robertson and Gary Starkweather. *Toward characterizing the productivity benefits of very large displays.* In *Proceedings of INTERACT*, 2003.

Gable et al., 2015. Thomas M. Gable, Bruce N. Walker and Andrew S. Amontree. *Investigating a new display format for CarPlay to decrease impact of mode change inputs.* In *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2015.

Hamilton et al., 2014. Peter Hamilton and Daniel J. Wigdor. *Conductor: Enabling and Understanding Cross-Device Interaction.* In *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 2014.

Nielsen et al., 2014. Heidi Selmer Nielsen, Marius Pallisgaard Olsen, Mikael B. Skov and Jesper Kjeldskov. *JuxtaPinch: Exploring Multi-device Interaction in Collocated Photo Sharing.* In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, MobileHCI '14, p. 183–192. ACM, 2014.

Pierce et al., 2008. Jeffrey S. Pierce and Jeffrey Nichols. *An Infrastructure for Extending Applications' User Experiences Across Multiple Personal Devices.* In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, p. 101–110. ACM, 2008.

Rädle et al., 2014. Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer and Yvonne Rogers. *HuddleLamp: Spatially-aware mobile displays for ad-hoc around-the-table collaboration.* In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, p. 45–54. ACM, 2014.

Reda et al., 2015. Khairi Reda, Andrew E. Johnson, Michael E. Papka and Jason Leigh. *Effects of display size and resolution on user behavior and insight acquisition in visual exploration.* In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.

Schmidt et al., 2012. Dominik Schmidt, Julian Seifert, Enrico Rukzio and Hans Gellersen. *A Cross-device Interaction Style for Mobiles and Surfaces.* In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, p. 318–327. ACM, 2012.

Seyed et al., 2013. Teddy Seyed, Chris Burns, Mario Costa Sousa and Frank Maurer. *From small screens to big displays: understanding interaction in multi-display environments.* In *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion*, 2013.

Skov et al., 2015. Mikael B. Skov, Jesper Kjeldskov, Jeni Paay, Heidi P. Jensen and Marius P. Olsen. *Investigating Cross-Device Interaction Techniques: A Case of Card Playing on Handhelds and Tablets.* In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, OzCHI '15. ACM, 2015.

Sørensen et al., 2014. Henrik Sørensen, Dimitrios Raptis, Jesper Kjeldskov and Mikael B. Skov. *The 4C framework: principles of interaction in digital ecosystems.* In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, p. 87–97. ACM, 2014.

Vatavu et al., 2014. Radu-Daniel Vatavu and Matei Mancas. *Visual attention measures for multi-screen TV.* In *Proceedings of the 2014 ACM international conference on Interactive experiences for TV and online video.* ACM, 2014.

Appendix

Project Disc A

The disc on this page contains for following:

- This report as PDF.
- The appendix report as PDF.
- Video and screen recordings from the user study.
- Source code of the project.
- Compiled application
- Link to Splashtop WiFi Display
- A link to the video demonstration of the system.