FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Interaction Techniques for Manually Rearranging Images on Mobile Devices

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Media and Human-Centered Computing

eingereicht von

Birgit Chmelar, BSc
Matrikelnummer 01227330

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Assoc. Prof. Dr. Dipl.-Ing. Hilda Tellioğlu
Mitwirkung: Mag.rer.soc.oec. Dr.rer.soc.oec. Roman Ganhör

Wien, 29. April 2019

Birgit Chmelar   Hilda Tellioğlu

FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Interaction Techniques for Manually Rearranging Images on Mobile Devices

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Media and Human-Centered Computing

by

Birgit Chmelar, BSc
Registration Number 01227330

to the Faculty of Informatics

at the TU Wien

Advisor: Assoc. Prof. Dr. Dipl.-Ing. Hilda Tellioğlu
Assistance: Mag.rer.soc.oec. Dr.rer.soc.oec. Roman Ganhör

Vienna, 29th April, 2019

Birgit Chmelar   Hilda Tellioğlu

Erklärung zur Verfassung der Arbeit

Birgit Chmelar, BSc
Szabogasse 323, 3040 Neulengbach

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 29. April 2019

Birgit Chmelar

Acknowledgements

I‘d like to thank Hilda Tellioğlu and Roman Ganhör for their support and for giving me the opportunity to work on this interesting topic. I also want to thank Peter Fikar for his advice.

My thanks also go to each participant for taking part in the focus group, the interviews or the usability tests.

Special thanks to my numerous proof readers and friends for their time and their encouraging words. It would not have been possible without your companionship throughout the period of study.

Last but not least, I want to thank my parents for everything.

Kurzfassung

“Ein Bild sagt mehr als tausend Worte” ist ein Sprichwort, das die Ausdruckskraft von Bildern hervorheben soll. Bilder können Erinnerungen wachrufen oder Geschichten erzählen, die reale Ereignisse widerspiegeln oder frei erfunden sind. Fotos werden gerne verschickt oder hergezeigt, um diese Geschichte weiterzugeben und um visuelle Eindrücke zu vermitteln. Oft befinden sich Fotos schon am Smartphone und können von dort unkompliziert verschickt werden. Es ist naheliegend, Diashows aus ausgesuchten Fotos von Highlights zu präsentieren, die nicht zwangsläufig in chronologischer Reihenfolge sein müssen. Möglichkeiten, um Bilder nach Metadaten zu sortieren, existieren im Überfluss in Galerien, jedoch mangelt es an Möglichkeiten, Bilder manuell anzuordnen. Sortieralgorithmen, die mit Metadaten arbeiten, können nicht antizipieren, wie Bilder für eine Geschichte angeordnet werden müssen. Da es an Werkzeugen zum manuellen Anordnen fehlt, ist auch nicht viel darüber bekannt, welche Anforderungen für diese Aufgabenstellung vorhanden sind. Die vorliegende Arbeit widmet sich dieser Problematik. Initial gesammelte Designansätze werden aufbereitet und analysiert, Ergebnisse der Analyse werden als Basis für eine Fokusgruppe verwendet. Sowohl die gesammelten Designansätze als auch die Fokusgruppe dienen zur Ideenfindung für ein vorläufiges Konzept. Da handelsübliche Smartphones die Zielplattform darstellen, wird das Konzept durch Screenshots auf einem Smartphone präsentiert. Dabei werden die Funktionen simuliert, die für manuelles Sortieren benötigt werden oder von Vorteil sein können. Interviews mit Leuten aus dem Bereich der Videobearbeitung werden geführt, um deren Expertenwissen in den Designprozess einfließen zu lassen. Ein Teil des Interviews widmet sich den Erfahrungen der Interviewpartner, der andere Teil des Interviews wird dafür aufgewendet, das zuvor entwickelte Konzept zu diskutieren. Ein Prototyp wird auf der Basis der angewandten Methoden entwickelt und während des Implementierungsprozesses iterativ getestet. Das Resultat der Usability Tests ist überwiegend positiv, der Prototyp wird von Teilnehmern als erlernbar und schnell bedienbar bezeichnet. Durch das Anwenden einer user-zentrierten Herangehensweise konnten essenzielle Anforderungen erhoben werden. Mit der Einbeziehung zukünftiger BenutzerInnen wurden zentrale Konzepte ausgearbeitet, die diese Anforderungen erfüllen, diese beinhalten das Verschieben von Bildern, eine Mehrfachauswahl, das Erstellen von Gruppen, eine Vollbildansicht, das Aussortieren von Bildern, eine Hilfestellung, ein Positionsmodus und eine Feinsortierung. Diese Konzepte wurden im Prototypen implementiert.

Abstract

Images are an expressive medium to remember experiences or to tell a story, which may or may not be purely fictional. People commonly store a great amount of images on their smartphones. The device is often used for showing or sending images to share visual impressions, therefore mobile devices seem to be an interesting platform for creating picture stories and to share them with others (e.g., in form of a slideshow). Usually only a selection of images is shared with others to depict some highlights of experiences made, such as a holiday travel. The original order of stored images does not necessarily need to fit the intended order of the narrative. Although meta-based ordering exists in abundance in gallery apps, there is a lack of tools which offer users to create an arrangement manually according to the story they would like to tell. Due to this lack of available solutions little is known about requirements for a technology, which handles such a task. The work at hand deals with this issue and presents the results of scientific research regarding this topic. Collected data comprises the results of literature review and gathered design suggestions were analyzed, serving as a base for a focus group in a further step. This provides the foundation for the ideation of a mockup, which simulates functionalities to achieve manual rearrangement of images on a mobile device according to the user's wishes. Interviews with video editing experts are conducted to explore how professional requirements could be met and how they may differ from the one's of common users. Furthermore, a part of the interview is dedicated to revive the mockup for discussion. The prototype was iteratively tested during the development, to identify strengths and weaknesses of the proposed implementation and gather input for the next iteration. The outcome of the usability tests is positive, according to the participants the prototype is learnable and quick to use, which is in accordance with the identified requirements of such a technology. The key contribution of this work lies in the specification and detailed description of a technological solution for the manual rearrangement of images. Interaction techniques such as image moving, multiple selection, collapsing of images, fullscreen view, trash view, position mode, finesort mode and a help view meet the identified requirements and are incorporated in the prototype. User-centered approaches aided the design and implementation of interaction techniques for the task at hand.

Contents

CHAPTER 1

Introduction

Oxford Dictionaries define storytelling as "*The activity of telling or writing stories*" [76]. Bernard [13] mentions the importance of the interplay of time and plot for the story. She introduces the term story as follows "*A story is the narrative, or telling, of an event or series of events, [...]*" [13], furthermore she adds the aspect of emotion "*It engages the audience on an emotional and intellectual level, motivating viewers to want to know what happens next*" [13]. Images can be more expressive in some situations than words or text, they can complement a narrative. "An event or series of events" can be depicted by images. Telling a story, needs the images to be in a specific order according to the narrative.

Due to the popularity of smartphones, a device for taking pictures is at hand in many situations. Events or remarkable impressions can be captured on photos, to be remembered, to share them with others or to use them to illustrate experiences when telling a story. Smartphones are powerful tools for a wide variety of tasks, due to sensors that are integrated, the amount of available applications and the openness for third parties to develop further applications [88][49]. Mobile phones evolved from their original use of texting and calling [9] to a companion in everyday life for a majority of people [5][60]. Taking pictures in every situation also proved to be a popular use of smartphones. One of the major motivations for making photos and videos is to share them. This is also highlighted by Kindber et al. [54] in their work on investigating the reasons why people take videos with their camera phones.

1.1 Problem Statement

Pictures can be arranged in a specific order, to communicate a story. Narratives emerge from the imagination of the author. Therefore only the author knows the order, in which to arrange images and no algorithm can predict it. The images may not be taken in the order the author wants to use them in the story, or may even exclude some of them completely. The story may also consist of images, which were not taken by the author. The story itself might even be purley fictional. In any case, if the photos shall match a specific story the author has to arrange them accordingly. Only the author has a vision of which pictures to use in which order and which ones are not important to convey a message. The order can not be done with automated sorting of images based on low-level features, like color or shape, and also not by meta data like file type or capture date or camera model for photos. While automatisms proved useful in many areas of applications, some tasks have to be executed manually. Some associations made by human minds may be impossible to predict by a machine, no matter how flexible it is. Some use-cases may also be highly specific such as gathering photos for assigning them to a construction blueprint or similar.

Videos are also an appropriate medium to tell stories. They are basically a sequence of pictures, that create the effect of motion if a specific frame rate is hit [40]. At a rate of 24 frames per second (fps) the viewer experiences the so called *short-range apparent motion*. Humans receive fluency by interpolating missing information between images that are slightly different. Since the first film in 1895 [21] there was an big evolution of technologies, techniques and plots. The rise of digital technologies introduced new possibilities. It led to tools, formats and codecs, which had an impact on the production of films. Regardless of the medium of recording digital images or videos, media assets can be edited and combined to shape a narrative. Digital footage transformed linear film editing to non-linear editing. In cases where editing took place with videotapes it was not possible to jump to sequences from the end of the tape without forwarding the tape. Cutting a video actually involved slicing the footage and glueing it together in a different sequence. Digital media files offer *random access*, it is possible to jump to arbitrary points in the media file with just a click. This justifies the term non-linear, because the access to spots in the video is not linear [40][70]. Nonlinear editing systems (NLEs) are used in the post-production of filmmaking, to arrange media files. Popular software in the field of video editing are for example Apple's Final Cut Pro [1] and Adobe Premiere Pro [2]. The interface of the video editing software always contains some specific elements, independent of the manufacturer, although the names of those views vary. One element is a timeline which enables the editor to organize the order of video clips. This highlights the importance of the sequence of videos, and the commonality that is shared with the topic of this work, the arrangement of media files to tell a story.

NLE softwares are mighty tools for video editing, but they are too complex for the purpose of this work, offering a lot of features for professional purposes. Furthermore the aim of this work is interested especially in the interaction and interface part of arranging images on mobile phones. Photos usually already exist on the device and enables to compose images into a story en route. The task is cumbersome on smartphones for two major reasons. They have a limited display size which makes it hard to maintain an overview over a big amount of images. The control of touchscreens by finger makes it

[1] `https://www.apple.com/final-cut-pro/`, last accessed: 03/04/2018
[2] `https://www.adobe.com/products/premiere.html`, last accessed: 03/04/2018

hard or impossible to precisely interact with small elements on the screen. There were no studies found in scientific literature regarding the design of interactions for manual organizing and rearranging images on smartphones.

1.2 Aim of the Work

The aim of this work is to identify and elaborate design requirements for efficient interaction techniques for manually rearranging images on mobile devices. Arranging images requires to search and find images in photo collections, which is sometimes cumbersome due to a vast amount of images. Pictures are shown as thumbnails, which are small versions of the images with reduced quality. User research is conducted to meet demands, which serve as a base to develop a mobile application prototype. The research question, guiding this work is "How can human-computer interaction (HCI) support the arrangement of images?". This question furthermore implies the following questions: "How are images ordered on digital devices for image browsing?" and "How is direct user manipulation integrated for rearranging images?".

1.3 Methodology

The methodological approach encompasses the review and analysis of over 100 design concepts specifically dedicated to manual sorting and a categorization of these concepts to provide the building blocks for a focus group with interaction designers. The focus group is supposed to raise qualitative outcomes and refine design requirements. Furthermore, the focus group should serve the ideation process. Based on the design concepts and the focus group an initial mockup will be created. Interviews with video editing experts are conducted to allow for insights into a related profession. One part of each interview will build up on the mockup, to discuss its current state and functionality. The outcomes of all methods are then merged into a final concept. Eventually, the concept is implemented for mobile phones and evaluated. The development will be iterative, user tests are conducted at the end of each of three milestones to incorporate user feedback in the final product.

1.4 Structure of the Work

The structure of the work is as follows: related work will be presented in Chapter 2. Terms and algorithms for image retrieval are introduced, which are important for the understanding of succeeding related work. Afterwards input modalities for digital devices are discussed, especially mobile phones and touch input are covered. Digital image collections and image browsing is introduced, before the topic of video editing is illustrated. In Chapter 3 the methods used in the thesis are described, beginning with the introduction of the term User-Centered Design and a presentation of applied methods, including interviews, a focus group, iterative prototyping and usability tests. Chapter 4 explains the actual implementation done in the course of this thesis. It addresses setup, deployment and analysis of each method and describes the process in detail. This also includes the specification of the final prototype. Chapter 5 discusses the implementation and the overall approach of this thesis critically. Important outcomes are highlighted, beneficial results and issues are examined and starting points for future work are proposed. How the research questions have been addressed throughout the thesis will be explicitly discussed in this chapter. Chapter 6 concludes the work by summarizing the most important aspects.

CHAPTER 2

Related Work

This section presents topics regarding image browsing, video browsing and video editing. Fundamentals of interaction design techniques will be presented and examples of the three topics will be addressed, including a selection of input devices and gestures for touchscreens. Relevant terms used in the examples will be defined to clarify their meaning. Some terms are in close relation to computer vision methods, which are used for the management of large image databases or pattern recognition. This is of interest, because some of the presented prototypes include such mechanisms. Even though this thesis does not include such mechanisms, they are discussed to provide a better understanding of the presented examples. Image and video browsing are covered from different perspectives, including personal computer applications and smartphone applications.

2.1 Terminology

This section explains and defines some relevant terms and algorithms such as *"browsing"* and *"searching"*. Afterwards a brief introduction into image retrieval is given. Methods related to computer vision, even though they are not integrated in the prototype, serve two purposes. Firstly, the difference between the aim of the work and the state of the art regarding elaborated algorithms is shown. Secondly, it will help to understand the examples presented later in this chapter.

2.1.1 Searching vs. Browsing

The reviewed literature often does not make a distinction between the terms *search* and *browse*. In this work, the distinction will be useful in some cases, therefore both terms are be defined here. Oxford Dictionaries defines the term "search" as follows: *"Try to find something by looking or otherwise seeking carefully and thoroughly"* [76]. Browsing, in contrast, is described as *"Scan through a text, website, or collection of data to gain an* *impression of the contents"* [76]. Thus, browsing is more casually, while searching is more targeted. Therefore exploration of data can be searching, browsing or a combination of both. Datta et al. [22] discuss image search as well and introduce three types: *"(1) search by association, where there is no clear intent at a picture, but instead the search proceeds by iteratively refined browsing; (2) aimed search, where a specific picture is sought; and (3) category search, where a single picture representative of a semantic class is sought, for example, to illustrate a paragraph of text, as introduced in Cox et al. [2000]"* [22, p. 5:6]. The first and third type refer to the herein defined term of "browsing", the second refers to "searching".

2.1.2 Clustering and Categorization

Some browsing tools incorporate clustering or categorization to improve efficiency. Categorization is based on prior known classes that the items can be assigned to. Clustering groups items based on features which have a high similarity within the group. Categorization therefore belongs to the category of supervised learning, in which training data is labeled with known classes. Clustering belongs to the category of unsupervised learning, without prior knowledge about the resulting clusters. Several types of data can be used for clustering and categorization as well as for searching and browsing. One group is *low-level image features*, such as color, shape or texture information. The literature research showed various projects that incorporated color information for ordering images. *Semantic* information can be manually added or extracted from images automatically.

2.1.3 File Organization

Organization has two impacts on this work. On the one hand, organization of images is important to retrieve images of interest. On the other hand, rearranging images to obtain a specific order is also a part of organization. Studies about human organization with the background to develop ICTs is not novel, Malone [63] conducted a study in which he interviewed people at their workplace to find insights about their organization strategies. He presented three stages of finding information: *Creating Classification*, *Classifying Information* and *Retrieving Information*. The first is concerned with hierarchical structures. Malone suggests to make the creation of multileveled classification in computer systems simple. To ease the process of classifying information, he states three non-exclusive choices: 1) Allowing multiple categories for one file, 2) allowing to defer the classification process and 3) providing an option for automatic classification. For the last step, the retrieval of information, it is recommended to enable searching for more than one property if a specific file is of interest.

2.1.4 Multimedia Database Management Systems and Content-Based Image Retrieval

This excursion into the field of Multimedia Database Management Systems (MMDBMS) and Content-Based Image Retrieval (CBIR) deals with the organization of information, especially with images. Since it is not the actual topic of this work, MMDBMS and CBIRs will not be discussed in depth, but on a level to familiarize the reader with concepts that are part of the presented related work. MMDBMSs have the purpose of organizing multimedia data. They are able to maintain text, images, graphic objects, animation sequences, video, audio and composite multimedia [2]. Managing information does not only include storage but also retrieval thereof. Retrieving images is more challenging in comparison to retrieving text, since images usually lack descriptions. Therefore other approaches are deployed for the image data type. CBIR is concerned with search processes based on the content of images, rather than with their textual metadata. Storing images in a structured form is called indexing. Bashir et al. [11] discuss 3 levels of CBIRs. Level 1 works on low-level features. Level 2 incorporates computer vision in the form of processing semantic information, operating on the identity of objects within images. Level 3 is based on artificial intelligence. The purpose and meaning of objects, which can be deduced with Level 2 techniques, are used for retrieval. MMDBMSs enable browsing and searching of content. A CBIR system receives a query from the user and calculates a matching score internally. This depends on a similarity metric that can be applied on features that are stored in the database after being extracted from the image data. The CBIR presents the results ranked by similarity to the user's query [92]. Zhang also mentions the use of relevance feedback for improving the search results. Image retrieval offers different query types. One option is "textual", using either keywords or free text for describing and retrieving images. *Annotations* label images with semantic information [89]. This can be accomplished manually or automatically. *Tagging* is one kind of annotation, it allows the user to choose keywords for annotations to describe and organize items. Tags can be used for searching or browsing through image collections. "Query by Example" and browsing are two other methods for image retrieval [34].

2.2 Input Modalities in Human Computer Interaction

This section is concerned with the interaction between humans and machines. It focuses on the input, covering input devices and input modalities that are relevant for this thesis. This includes examples for image and video browsing, input devices, mobile devices, touchscreens, gestures, and technical terms.

2.2.1 Input Devices

Input is a term for human actions that serve to control a computer. Figure 2.1 depicts the interface between human and machine. The human input changes the internal state of the machine, which produces a certain output as reaction [62]. "Interaction" and "Technology" are two topics, that are covered by MacKenzie. In MacKenzie's work, technology is concerned with the physical properties, which enable the interaction.

Jacko [45] names a number of characteristics for input technologies. The authors distinguish between *absolute input devices* and *relative input devices*. Those are characteristics which define the type of the mapping between input and output. Absolute input devices

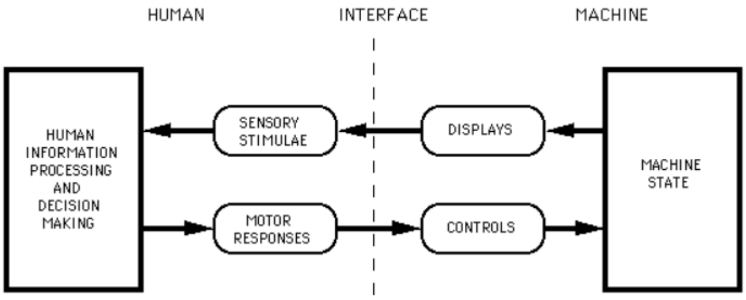


Figure 2.1: Interdependency between human and machine. The human interacts with the machine through an interface [62].

		Number of Dimensions							
		1		2				3	
Property Sensed	Position	Rotary Pot	Sliding Pot	Tablet & Puck	Tablet & Stylus	Light Pen	Floating Joystick	3D Joystick	M
				Touch Tablet		Touch Screen			T
	Motion	Continuous Rotary Pot	Treadmill	Mouse			Trackball	3D Trackball	M
			Ferinstat				X/Y Pad		T
	Pressure	Torque Sensor					Isometric Joystick		T

Figure 2.2: Table of input devices classified by several factors of input modalities [16].

define a point on the screen, they are independent of motion and are also called position sensing devices. Visual feedback like a cursor is necessary to show the position on the screen. Relative input devices depend on the motion of an input device. Furthermore, the number of dimension (or degree of freedom) and indirect versus direct interaction shall be mentioned. In the case of direct pointing devices the input happens physically on the screen, while indirect interaction has to be mapped from a pointing device to a point on the screen.

Buxton [16] created a table with categorizations of a number of input devices. Devices are split in two main sensed dimensions: the property that is sensed, which is either pressure, motion or position, and the number of dimensions, which are either 1, 2 or 3. The sensed property is further divided into the mechanism of how the properties are detected: either directly, by touch - “T”, or mediated by mechanical devices “M” (Figure 2.2). The table only contains continuous input, in contrast to discrete input. Devices are categorized by demanded motoric skills. Cells which are vertically divided by dotted

lines have similar motoric requirements. The input device table is not exhaustive, but sufficient for the scope of this thesis.
Schoeffmann et al. categorized reviewed papers regarding video interaction tools in three categories: 1.) keyboard and mouse, 2.) touch screen and 3.) environmental. The first is described as traditional, like desktop or laptop. Touchscreens are smartphones or tablets and environmental input methods use physical objects on a table or in a room [81]. The following four input modalities are used in one or more examples for image browsing, video browsing or video editing.

Mouse. The computer mouse is an indirect pointing device with a relative mapping. It has two degrees of freedom: motion in the x and the y direction. A mouse usually has at least two hardware buttons and a scroll wheel to interact with graphical user interfaces. Alternatives are the touchpad and the trackball, which are often part of laptops.

Touchscreen. Touchscreens can be part of devices like smartphones, tabletops or tablets. Touchscreens also belong to the category of pointing devices. In contrast to indirect tablets like graphic tablets or digitized tablets, touchscreens have a direct input. However, graphic tablets and digitized tablets are often operated with a stylus, which can also be the case for touchscreens. This work focuses on smartphones, tablets and tabletop devices, which are equipped with touchscreens that can be operated by user contact, usually with one or more fingers.

Keyboard. The standard input devices for desktop computers are mouse and keyboard. Unlike the mouse, the keyboard is no pointing device. It serves the purpose to enter text for different applications or to navigate with the arrow keys. Most reviewed papers use only pointing devices or enable optional keyboard input to interact with the prototype. Only one of the presented papers explicitly requires the keyboard for text search.

Motion. Motion detection can be facilitated by gyroscopes or by tracking the human with a camera and processing the image data. One example will be given, which compares two different motion detecting methods (one with a camera, one with a motion tracker) and an interface with buttons for image browsing. Common handheld devices have a built in gyroscope and other sensors to track motion for a variety of applications [88].

2.2.2 Gestures for Touchscreens

This section describes various kinds of touch input for human-computer interaction. Mobile phones, which are equipped with touchscreens, can be operated by gestures. Touchscreens are coupled to touch-sensitive tablets. Their input modalities are categorized as direct input [45]. Gesture recognition is one research area of Visual-Based HCI [51]. In general, the term gesture can be used in a broader definition, taking into account the whole body or, at a narrow definition, only hand gestures [93](facilitated by gloves) [82] (detected in a video stream). In this thesis the term gesture refers to direct input on devices with touchscreens.

Minsky developed a prototype in 1984 to recognize single-finger gestures on an engineered touch-sensitive monitor [66]. The position and pressure of a touch point could be accessed by the system, fulfilling the purpose of a paint program. One part of the prototype was a gesture parser recognizer, which matched detected touches to one of three gesture types: selection, move and a path gesture. Those gestures facilitated the manipulation of displayed objects on the screen. Other applications were also implemented for experimentation on the system.

Smartphones are shipped with operating systems that handle touch gesture input. Some gestures established for touch devices are independent of the operating system. Although gestures might be called differently by various companies, they work the same way. Common touch gestures, which are enumerated in the documentations and guidelines for Android[1], iOS[2] and Windows[3] are discussed and Table 2.1 describes common gestures. The table presents terms that are used in one or more of the previously mentioned operating systems of mobile phones. Different names for the same gesture, that vary between operating systems are listed.

Android refers to an external website holding information about "Material Design", which is introduced as *"Material Design is a visual language that sythesizes the classic principles of good design with the innovation of technology and science"* [30]. Android suggests to follow the "Material Design" guidelines for the development of android applications. The website categorizes three types of gestures: navigational gestures, action gestures and transform gestures.

Navigational gestures *"... help users to move through a product easily. They supplement other explicit input methods, like buttons and navigation compontens"* [58]. Five gestures belong to that category:

- Tap
- Scroll and pan
- Drag
- Swipe
- Pinch

Action gestures are defined to *"... perform actions or provide shortcuts for completing actions"* [58]. The following three gestures are action gestures:

[1] `https://www.android.com`, last accessed: 03/31/2019
[2] `https://www.apple.com/at/ios/ios-12/`, last accessed: 03/31/2019
[3] `https://www.microsoft.com/de-de/windows`, last accessed: 03/31/2019

- Tap
- Long press
- Swipe

Transform gestures are used to transforming size, position or rotation of elements, that are shown on the screen. Four gestures are enumerated in this category:

- Double tap
- Pinch
- Compound gestures
- Pick up and move

Microsoft distinguishes between static and manipulation gestures. Static gestures are “tap” and “press and hold”. Manipulation gestures are dynamic interactions, namely “slide”, “swipe”, “turn”, “pinch” and “stretch” [20].

Apple describe their standard gestures in their function. Therefore they do not explain how gestures are performed, but what effect they have in the operating system. They mention no categorization of the presented gestures [44].

In contrast to gestures which are established in the use of smartphones, developers are able to implement individual gestures. The company “Ideum” [4] developed a framework called GestureWorks[5], which is capable of detecting more sophisticated multi-touch gestures. Figure 2.3 shows an excerpt of their poster which depicts gestures that can be detected by the framework.

Apart from the gestures described in this section, touchscreens can be exploited in a different way for interaction possibilities. Bezel Swipe [77] is a method for selecting, cutting, copying and pasting of items on a smartphone screen. Actions of the Bezel Swipe start at the so called “bezel”, which is the touch-insensitive border of a display. Hereupon small bars are arranged on the touch-sensitive screen. They are activated when touched, which starts the selection mode, that can be imagined as a drag from the edge of the screen to an aim on the display. Each bar is colored differently and represents a different mode. For example, if text is selected, then there are two different bars for marking the start and the end of the selection. Figure 2.4 depicts three use cases of Bezel Swipe.

Another example of alternative interaction methods are “hot corners” or “active corners”. These are the corners of a screen with a hidden functionality. They are activated by specific actions. Avsar et al. [7] integrated hot corners in user interfaces of flight management systems for controlling the radio frequency. The interface shows a map with symbols for interactive elements. The pilot can alter the frequency by dragging and dropping a station, which is an interactive element, into the corner.

[4] `https://ideum.com/`, last accessed: 04/26/2018

[5] `http://gestureworks.com/`, last accessed: 04/26/18

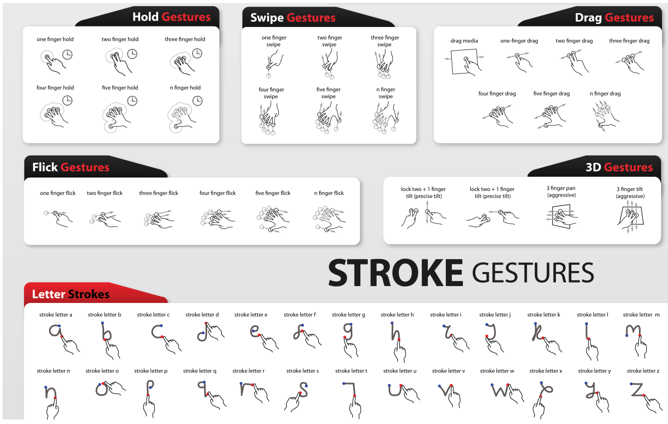


Figure 2.3: An excerpt of gestures, which are implemented in Ideums framework gestureworks [43].

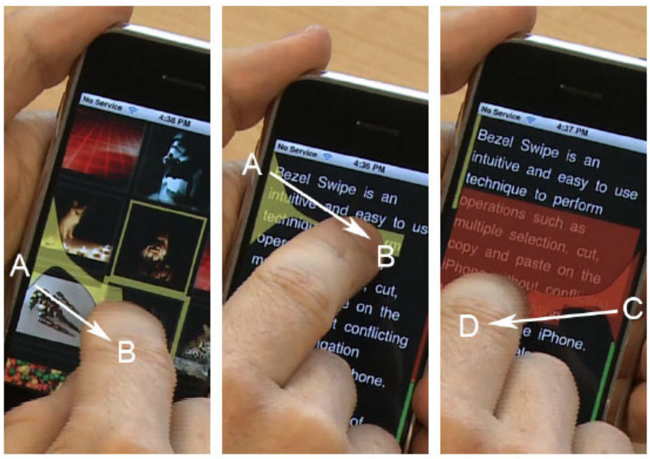


Figure 2.4: Use cases of Bezel Swipe. Left: swiping from A to B selects images. Middle: swiping from A to B sets the start of a text selection. Right: swiping from C to D sets the end of a text selection [77].

Table 2.1: Summary of Touch Gestures

Name	Description
Tap	One finger touches the screen and lifts up [20].
Double Tap	Tapping the screen two times in quick succession.
Press and hold	One finger touches the screen and stays in place [20]. Also referred to as “long press” by Android or “touch and hold” by Apple.
Swipe	One or more fingers touch the screen and move in the same direction. UWP differentiates between slide and swipe. In their description, swipe is a move of a short distance. Here, swipe is not constrained.
Flick	Quick swipe in one direction. Possibly comparable with UWPs definition of a swipe.
Turn	Two or more fingers touch the screen and move in a clockwise or counter-clockwise arc [20].
Scroll and Pan	Users can slide surfaces vertically, horizontally, or omnidirectionally to move continuously through content [44]. Vertical and/or horizontal limitation refers to scrolling, panning to omnidirecitonal movement. The gesture corresponds to a swipe or a flick, with the difference, that the purpose of the gesture is defined for specific elements, on which the gesture is applied.
Pinch	Two or more fingers touch the screen and move closer together or farther apart. UWP differentiates between the directions the finger move. Pinch refers to move the finger closer together, stretch refers to move the fingers farther apart [20]. Here, no difference is made between the direction, pinch can mean both directions. Pinching is often used, for zooming in and out [44] [58].
Drag	Also called “pick up and move” by Android. Elements can be dragged from one place to another on the screen. Often implemented by tapping and holding an element, moving the finger to the other position and releasing to drop the element.

Meschtscherjakov et al. [65] apply active corners in a touchscreen interface of cars. Passengers can interact and exchange digital items (i.e. a photo) by sending them to another passenger, which is supposed to encourage the collaboration of the passengers. Each of the four corners of the tablet is mapped to a passenger in the car (i.e. the driver is assigned to the left upper corner). The passengers can use either a mobile phone or a built in touch screen. Sending an item to someone else happens by dragging and dropping the item into the corner associated with the respective passenger. Furthermore a card game based on the active corner principle was developed by the authors to evaluate the principle.

2.3 Digital Image Collections

The remaining part of this chapter is concerned with image and video browsing and video editing. This section presents approaches from scientific literature for image searching and browsing. Different ways of representing a high amount of images are shown, each way is accompanied by an example from literature.

2.3.1 Representation

This subsection provides a set of representations emerging from the literature search. Each category is represented by at least one example. The representation of a image collection is not exclusive, hence more than one representation or aspects of representations can be existent in one prototype. Like other categories the presented types of views are not exhaustive. Prototypes can implement more than one type of view or a combination of them.

Grid

Many authors refer to the grid as the standard method to arrange images. Some researchers adduce the grid to build up on it to enhance the efficiency to browse through images by adding further mechanics. Agrafo is a system by Mota et al. [67] that uses automatisms to improve the organization of photo collections of the user. Three views were implemented in the user interface (Figure 2.5) including a grid. The mechanics draw on metadata, low-level features and semantic information. Agrafo is intended to build groups of images, seen on top of the screenshots as stacks. Multiple groups can be open at once to build new groups and join groups or moving images between and within groups by drag and drop. Semi-automatic grouping happens interactively. Criterias can be chosen from a pool and adjusted according to their relative importance to perform the grouping. The slider interface can be seen in Figure 2.6. The highlighted slider on the right side allows to vary the level of similarity between images within the group. This option also influences the number of groups that will be created by the system.

Another project, DynamicMaps [55] arranges thumbnails in a grid to represent the image

(a) Agrafo grid interface. (b) Agrafo shuffled interface. (c) Agrafo stacked interface.

Figure 2.5: Screenshots of all three Agrafo interface types [67].

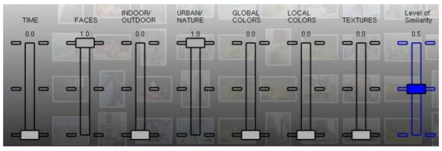


Figure 2.6: Agrafo grouping criteria selection [67].

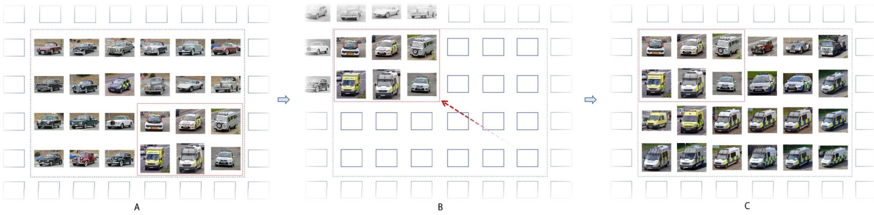


Figure 2.7: Panning in DynamicMaps [55].

collection. It works in a different manner and aims to form an infinite 2D grid. The grid, also referred to as the DynamicMap, is created continuously as the user interacts. Images are presented to the user in a grid, arranged in a way that considers the similarities between the images. Users may navigate into the direction they like best or that best matches a image of interest. As the user navigates into a specific direction the map will grow by loading and arranging images that are most similar to the already displayed images. Figure 2.7 shows the process in three steps. The selection of pictures that follows when the user browses is based on k-nearest neighbors, which, however, is based on the metrics of three image descriptors. The similarity score - in this case the euclidean distance - is precomputed for the images. The used parameters are the average color, the color histogram and the spatial envelope. The spatial envelope is also called the gist descriptor and relates to the context of a scene. The order in which images are retrieved to fill the map when navigating is determined by the number of reference images to deduce the similarity score. If more than one position is eligible, then the closest one to the direction of the user navigation is chosen. The order is depicted in Figure 2.8. The prototype enables zooming, which adjusts the similarity between images next to each other. Zooming out results in reducing the similarity, to provide a better overview. Images next to each other have a bigger similarity distance with decreasing zoom level. Zooming is shown in Figure 2.9. The user might double click on an image, which leads to the regeneration of the map with the highest similarity to the new seed image.

Hierarchy

Jing et al. [48] developed Google Image Swirl , a hierarchical image browser which uses both visual and semantic characteristics. The system ought to be more effective than common text querying, especially in cases where the image of interest would not be listed on top of the search result. Users proceed with an initial result set of their query by clicking on their favorite image which is shown in the shape of a hierarchical balloon

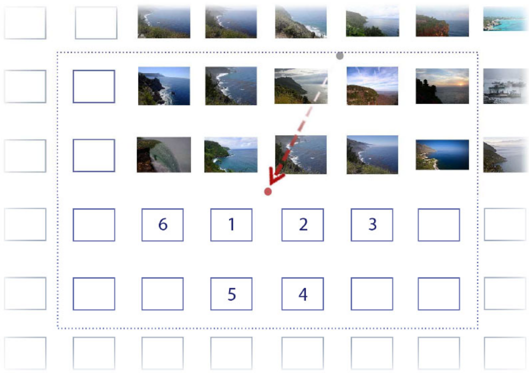


Figure 2.8: Order of “filling” new cells when user is panning [55].

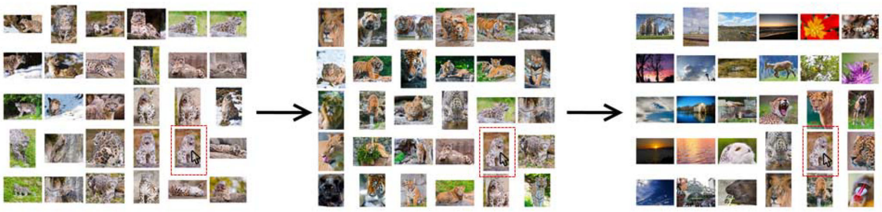


Figure 2.9: Zooming in DynamicMaps [55].

tree. Figure 2.10 shows steps of the interaction with Google Image Swirl. Dimensional reduction of several features (color, edge, texture, local features, etc.) is performed before computing the pairwise distance in the euclidean feature space. Clustering is applied to achieve the hierarchy for the current image.

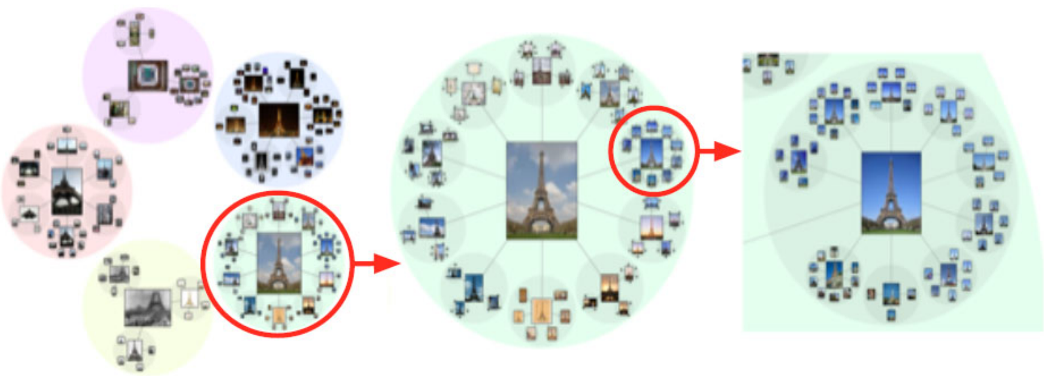


Figure 2.10: Tree representation of Google Image Swirl [48].

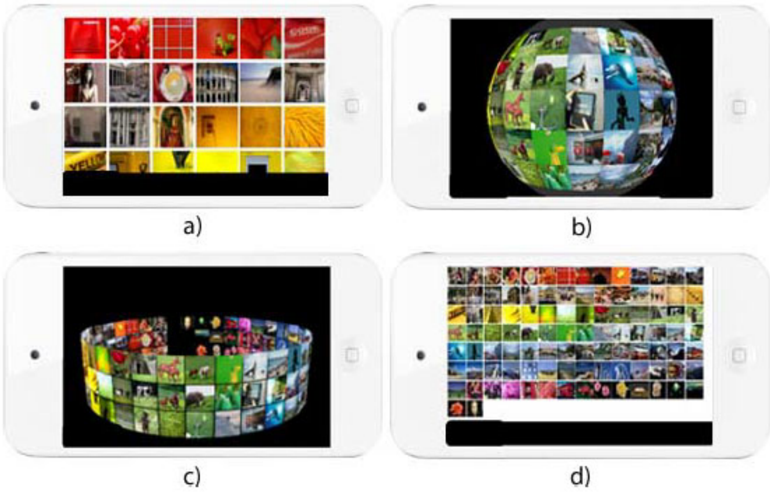


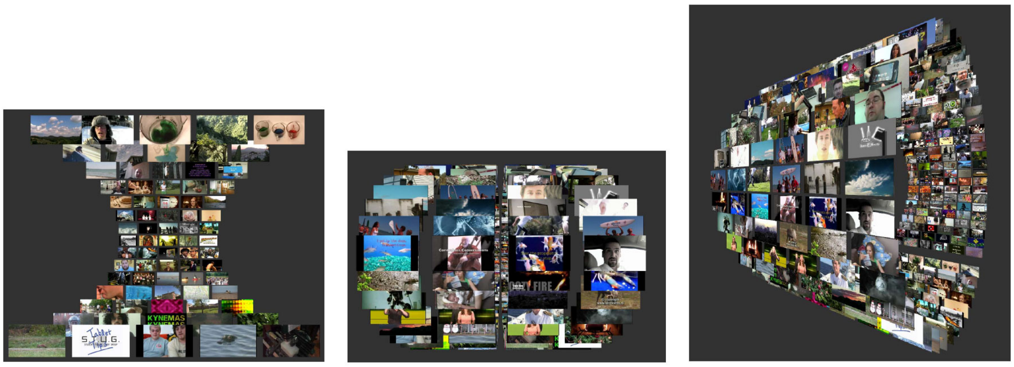
Figure 2.11: *Image browsing on small touchscreen devices: a) Grid; b) 3D-Globe; c) 3D-Ring; d) ImagePane* [38].

3D

Spheres and cylinders are often an metaphor images are arranged on. Representatives for both metaphors follow.

The *sphere*, or *globe*, is one of Hudelists [38] four approaches for image browsing and searching for mobile devices. The author uses color sorting in every approach, which is based on histogram of the HSV color space with 24 bins. Very dark and very bright images are positioned at the beginning and the end, respectively. A screenshot of each view is shown in Figure 2.11. The grid and the globe originated from an earlier study of Hudelist. The interaction on the grid involves swiping and dragging for scrolling. Images on the globe are arranged next to each other, depending on their HSV histograms. In this case swiping and dragging rotates and tilts the globe. Pinching enables the user to zoom in and out. In the ring prototype swiping vertically alternates between the original view and a zoomed view of the ring. Rotating the ring is accomplished by swiping horizontally. Double tapping changes the view to a zoomed view of the back of the ring, seen from the perspective of its center. In the ImagePane zooming is performed by double tapping, which also leads to focusing on the tapped spot. To leave the zoomed view the user double taps again anywhere. Gestures for nagivation are swiping and dragging, like in the grid prototype earlier mentioned.

Another example for 3D presentations in the area of image and video browsing is the *Cylindrical 3D Storyboard* by Schoeffmann and Boeszoermenyi [80]. The approach tries to overcome some of the limitations of conventional storyboards. The authors argue that grids with a large amount of images make it hard to visually compare images that are far apart. Other disadvantages mentioned are the lack of having an overview of all images as well as not having a detailed visualization. Furthermore, an unsuccessful search requires the user to start again from the beginning, which is directly next to the end. Images are horizontally positioned on a cylinder that consists of a number of "rings" which are placed next to each other. The radius and number of the rings are increasing with the number of images. The end is denoted by an empty row. Users can rotate the cylinder manually or let it rotate automatically. Furthermore users can zoom or change their perspective in the visualization, if they are interested in a specific thumbnail. The prototype provides the option to switch between three preassigned views, namely the center view, frontal view and right side view, which are depicted in Figure 2.12. The user has a better overview of items by peripheral perception. Those features are supposed to overcome the aforementioned limitations of conventional grid-like storyboards.



(a) Center view. (b) Frontal view. (c) Right sided view.

Figure 2.12: Three predefined views of the cylindrical 3D storyboard [80].

Timeline and Folders

Patel et al. [72] experimented with techniques for image searching and browsing on mobile devices. By gathering information from previous literature they noted three requierements of human search strategies. People need to be able to discover their already existing folder structures, they need to be able to search for labels in their folder structures, and they value viewing images according to absolute dates. Patel et al. tried to merge those requirements into an application. For temporal support the app includes a timeline. "Timeline Filtering" (TF) is a feature that can be applied to the timeline, allowing the user to define an upper and a lower boundary for dates. This leads to only displaying items which creation dates lie within the range. Figure 2.13 depicts three screenshots of the timeline on the left side of the screen and the filtering function next to it. Besides TF the app provides the option to access items directly that lie within a year or a month. This function, called "Timeline access" (T) by the authors, is applied by tapping on the corresponding area on the timeline. Months are labeled with their first letter in the timeline (from December to January) beneath the year. Images have a dark or a light grey background depending if they are within the currently selected folder (orange) or not, respectively. Additionally, the images in the app are arranged in

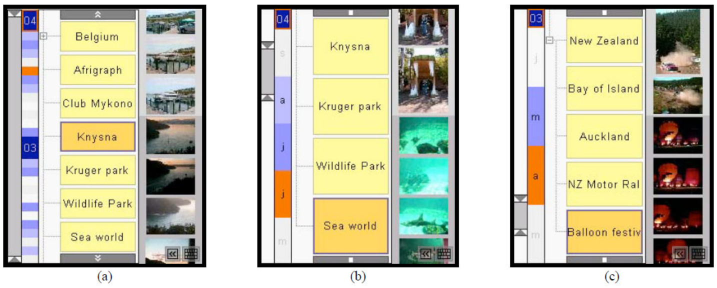


Figure 2.13: Timeline filtering [72].

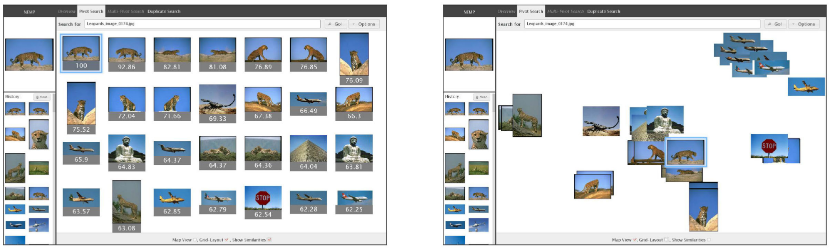


Figure 2.14: Grid-based and map-based view of the visual berrypicking map prototype [61].

hierarchical folders, similar to Microsoft Window Explorer. Hierarchical relations are visualized by small "+" symbols, which indicate the existence of subfolders and lines connected to corresponding subfolders if selected. Furthermore, a keyword search is implemented in the application.

Map-Based

An approach for browsing large image collections presented by Low et al. [61] is based on berrypicking, a strategy for information retrieval. A user refines As query step by step to improve the result. It is similar to DynamicMaps [55], which was introduced earlier, with the difference, that it also includes a map-based view. The authors build several maps of images, in which images are arranged based on color layout, scalable color, color- and edge histogram. This information is retrieved from the according MPEG-7 file. Dimensionality reduction is used, before a k-nearest neighbor algorithm is applied to a seed image. The resulting top k images are arranged in relation to their distance. A grid-like as well as a map-like layout were implemented in the prototype, seen Figure 2.14. As shown in the screenshots, it is possible that images are overlapping in the map view. This is avoided in the grid view, in which images are stored in the closest, empty cells. An image can be selected as new seed image in order to continue the search until the user is satisfied with an image in the result set. Every time a new seed image is selected, a new map is created overlapping the already existing map.

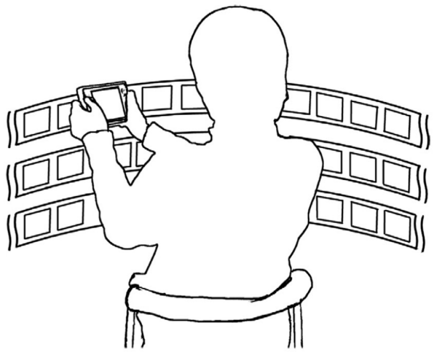


Figure 2.15: Layout of the motion based image browsing [91].

2.3.2 Image Browsing Examples

This section presents examples of image browsing, which do not match into one of the representation categories, but are nonetheless interesting. One of the examples uses motion input to move through the image collection, whereas another utilizes metadata for searching images in a collection. The last example in this section shows experimental visualization techniques for browsing through image collections.

Evaluation of Motion-Based Interaction for Mobile Devices: A Case Study on Image Browsing

In this section a motion interface that is facilitated by sensors in mobile devices is discussed. Three different devices are compared in a study. One of them only supports the input by buttons, the other two make use of motion data. One motion-based device is equipped with an accelerometer and a camera, the other one with a 6-DOF tracker to determine motion. Yim et al. [91] address this topic to circumvent disadvantages of the touch control of mobile devices. Fingers occlude the display when interacting with the touchscreen and the small size may complicate text entry. Pictures can be imagined to be arranged grid-like on a 3D cylinder. The user is located inside the cylinder and can move his or her device to view the images. The movement is transformed to show the correspondent image space. Figure 2.15 depicts the metaphor. The authors differ between two kinds of movement: *position control* and *rate control*. Position control maps the movement directly to the cursor displacement. A problem that might occur hereby is that the cursor position reaches the boundary of the physical workspace. In rate control however, the cursor movement velocity is affected, instead of the position. Yin et al. state that overshooting and precision are possible problems with rate control, which is the reason for combining position and rate control in their prototype. Figure 2.16 shows screenshots from experiments with users.

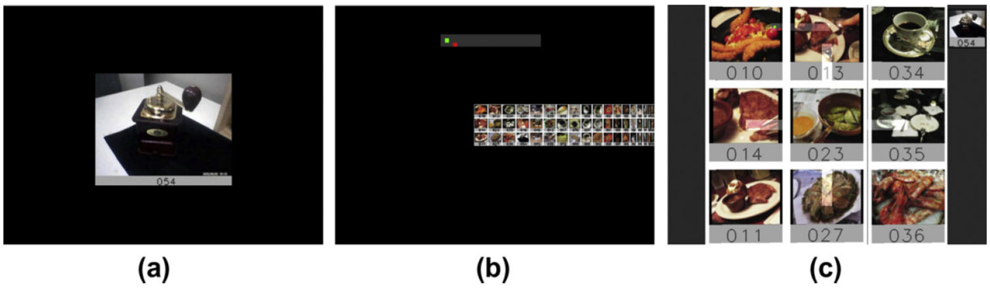


Figure 2.16: Views of the screen from the motion based image browsing prototype a) target image (to be found) b) location of target image shown in mini map c) task begins with an initial view [91].

Faceted Metadata for Image Search and Browsing

The standard input devices for desktop computers are mouse and keyboard. In the reviewed papers of prototypes keyboards are only used for entering search phrases. The mouse is also used in the prototype for selecting options by clicking. Yee et al. [90] developed an interface for keyword-based search, that makes use of faceted metadata. The metadata corresponds hierarchically ordered to categories of the images in the collection. Different characteristics of metadata are confronted in their work. Faceted metadata means that different sets of categories exist. Metadata can be hierarchical (in contrast to flat), and metadata can be single- or multivalued, which defines if more than one value may be assigned to an image. Metadata can be selected by clicking on pre-defined categories or by typing keywords for a query into a search field. The interface is designed in a way that works in three steps called *opening*, *middle game* and *endgame*. Each step is shown in a screenshot in Figure 2.17. Opening provides an overview over the whole collection as well as over the metadata facets. Facets are shown in different colors to make them easily distinguishable. The middlegame additionally offers the user to choose categories or to enter a search term. The difference is that those queries operate on the result step of the opening and accumulate chosen options to refine the result set. The currently selected facets are shown on the top of the screen. The endgame consists of a selected item that is shown along with the chosen facets that were also part in the middlegame.

Browsing Large Collections of Images Through Unconventional Visualization Techniques

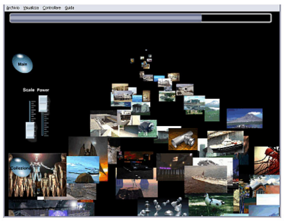
Porta [73] approached image browsing for people who do not know what they are looking for, with the implementation of seven different browsers. He criticizes that in conventional methods images are displayed in a grid which makes it hard to go through them quickly in cases of numerous images. The first approach is the *Elastic Image Browser*. Images are arranged in a grid, but in comparison to conventional methods, the user has the ability to scale images of the collection in vertical and horizontal direction independently,

(a) Opening. (b) Middlegame. (c) Endgame.

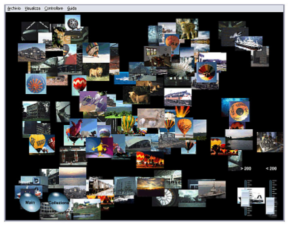
Figure 2.17: Steps of the faceted metadata search [90].

in order to show more images at once. The mouse is used to control parameters on the sliders like the shrinking percentage and the scrolling on the grid. Figure 2.18 shows all image browsers by the author. In the *Shot Display*, images are emitted from a point on the display and move to the bottom of the screen increasing their size in a perspective manner. Frequency and speed of emitted images can be adjusted by a sliders. It is also possible to control the scale of the images. Clicking on an image adds them to a collection for later use. The third prototype is the *Spot Display*. Images appear randomly distributed on the screen and are visible till another one appears on top, that occludes it. Again, the scale of the images can be adapted and clicking on an image leads to adding it to a collection. The *Cylinder Display* presents images randomly arranged on a virtual rotating cylinder. The surface of the cylinder can fit up to 200 images, and new sets can be retrieved by pressing the arrow keys. The rotation speed is controlled by the vertical mouse position. The closer it gets to the top screen edge the faster it rotates. Analogously, moving the cursor to the bottom screen edge slows the rotation down to zero. A slider offers the adaption of scaling the image size, clicking on an image saves it in the collection, like the previously introduced prototypes. Additionally, the user can choose to bring the selected image on top of the others and increase its size. The next approach is called the *Rotor Display*. It consists of four planes, each one containing one hundred images arranged on a grid. The planes are positioned in 90 degree angles to each other. Equally to the Cylindric display, the Rotor display rotates with a speed that can be controlled by the vertical position of the mouse cursor. Again, independent vertical and horizontal scaling can be controlled with sliders. The second to last presented prototype by Porta is the *Tornado Display*. It imitates a tornado of images, that move in this manner accordingly. Scale of images can be adapted by a slider, clicking adds it to a collection. Speed depends on the mouse position, based on the center of the screen. The last devised prototype is the *Tornado of Planes Display*. It also behaves in a tornado manner like the earlier approach, with the difference that the images do not swirl separately, but they are arranged on grids, corresponding to the "planes" in the name. Clicking adds images to the collection, scaling can be performed by varying parameters through sliders.

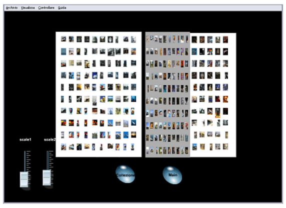
(a) Elastic image browser.



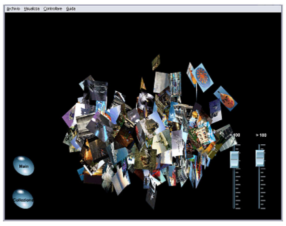
(b) Shot display.



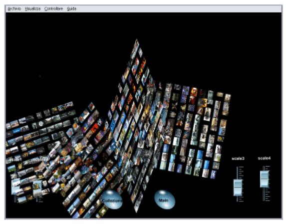
(c) Spot display.



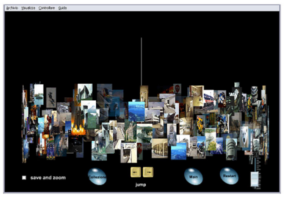
(d) Rotor display.



(e) Tornado display.



(f) Tornado of planes display.



(g) Cylinder display.

Figure 2.18: Browsing large collections of images through unconventional visualization techniques [73].

2.4 Digital Video

A video is a sequence of images, which have a specific order. The images in the context of a video are called frames. The frame rate defines the number of images per second. Fast iteration of these images causes the effect of movement. Frames expand images using the 4th dimension, time [46], and turn them into a time-dependent media. Therefore video editing involves arranging footage (raw, un-edited video material) in a specific order to create the final video. Jackson [46] names three steps in the digital video workflow: capture, edit and export. The author names *new media software genres* which are part or can be part of the video production process. These genres include *digital illustration*, *digital audio*, *digital imaging*, and *visual effects*. The aspects of sound and special effects in video editing are mentioned only for the sake of completeness, the focus of this section lies on the video material itself. Because videos consist of a sequence of images, it can be valuable to review them for the purpose of rearranging images. The literature on video interaction presents a variety of terms, which are defined below. Schoeffmann et al. [81] identified the following seven topics:

- **Video Annotation** - analogous to image annotations, this can also be applied to videos by annotating single frames or segments.
- **Video Browsing** deals with the exploration of videos. It makes use of video interaction mechanisms to enable searching within a video. Mechanisms which can be incorporated to facilitate browsing are video navigation, content querying, video annotation and video summarization.
- **Video Editing** is concerned with working with video footage and media files.
- **Video Navigation** refers to jumping to random access points within the video. Temporal positioning and context visualization are named as common methods for navigation.
- **Video Recommendation** systems evaluate which videos might be interesting for the user based on the users interaction behavior with videos.
- **Video Retrieval** is analogous to image retrieval. The purpose is to enable users to retrieve videos instead of photos. Interaction methods for videos are used to achieve a goal, which can be either to explore a high amount of data or known-item search.
- **Video Summarization** aims to present a summary of a video to the user by processing important information of the content.

Xiong et al. [87] approach video access from two perspectives. First, video browsing is top-down and suited for exploring videos without exactly knowing what to look for. Second, video retrieval is the second approach defined by Xiong et al. [87]. It is a bottom-up approach for known-item searching.

2.4.1 Video Summarization

As many of the following topics will build off of video summarization, it makes sense to start here. Ajmal et al. [3] surveyed the topic of video summarization by reviewing literature. Video Summarization techniques were hierarchically categorized, and the categories can be seen in Figure 2.19. *Feature Based* in video summarization works equally to images with features like color. Additionally video inherent features like motion and voice can be used. Ajmal et al. extracted seven subcategories of feature based summarization (Figure 2.19), which are further discussed in their paper [3]. Clustering can be applied for video summarization to group similar frames into segments, which can be represented by keyframes in the summary. Four subcategories of cluster based are summarized, which vary in the features and methods they use for clustering. *Event Based* summarization computes differences between frames and clustering to detect events within videos in order to present them in video skims or key frames for dynamic summary or static browsing, respectively. *Shot Selection Based* detects transitions in

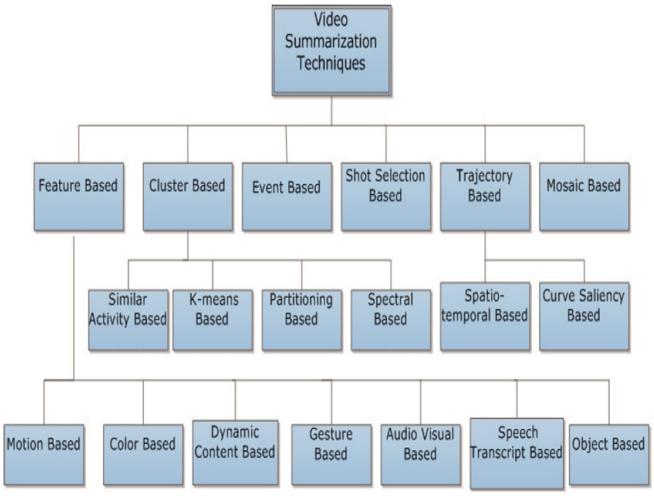


Figure 2.19: Summarization techniques according to Ajmal et al. [3].

videos. Shot detection, also called cut detection, attempts to split the videos up into parts that correspond to shots. A shot is a period, comprised of consecutive frames, that is recorded by a single camera [56]. The transition between shots can be abrupt (cut) or gradual (e.g. fading, curtain, etc.), which makes the detection a challenge. According to Aner and Kender [6] shots are one of three units of which video genres can be comprised, along with frames and scenes. According to Rui et al. [78] there are five categories for determining the boundaries between shots: *pixel based*, *statistics based*, *transform based*, *feature based* and *histogram based*. Ajmal et al.'s summarization include two more categories: *trajectory based* and *mosaic based*. *Trajectory Based* methods are well suited in situations in which the camera is fixed. It considers the movement of objects within a video to enable summarization. *Mosaic Based* techniques, also called *salient stills*, create panoramic images assembled of temporal changes of consecutive frames [85]. Transcoding video into images in this sense has to consider camera movements as well as object movements in a scene. Salient frames within a scene are projected into one reference frame.

2.4.2 Video Browsing

Christel et al. [18] worked on methods for skimming a video called "skims". This provides a shortened version of the video content for the user. The skims are produced by extracting certain parts of the video, either at regular intervals or through content-based analysis. The work includes prior work from the co-author Smith and Kanade [83], who also dealt with the creation of skim videos. They name *"fast-forward playback and skipping video frames at fixed intervals"* [83, p. 775] as an example for simple browsing techniques. Their method is more advanced and uses audio and video to extract significant information to create skims.

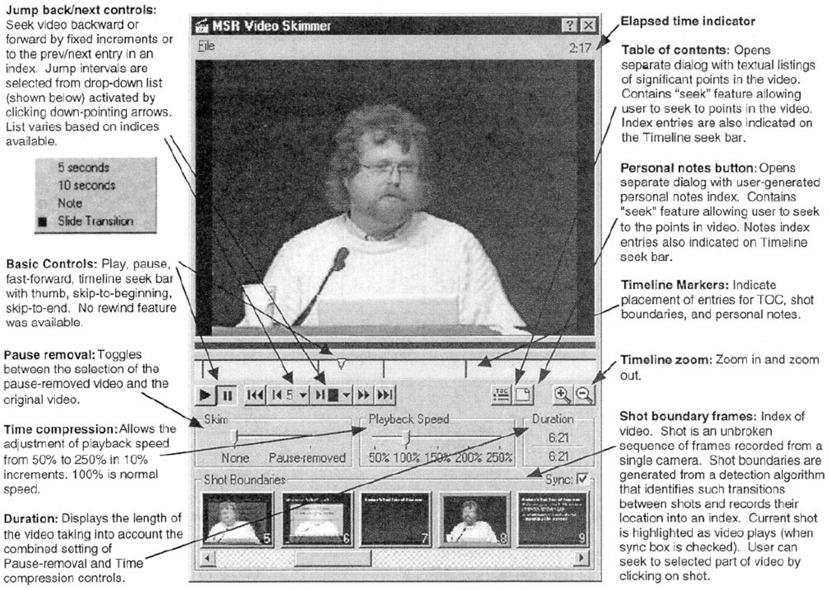


Figure 2.20: Browsing digital video, interface with advanced browsing techniques [59].

Li et al. [59] presented a tool for video browsing which included basic controls, like play, pause, skip-to-beginning, skip-to-end, fast-forward and seek. The authors expanded the browsing techniques of common systems with new features. This includes speed-up controls (time compression and pause removal), textual indices (table of contents and notes), visual indices (shot boundary frames and timeline markers) and jump controls (jump-back and jump-next). The interface including the details of the new features can be seen in Figure 2.20.

The following examples are essentially analogous to image browsing and chosen to co cover a variety of approaches. They differ from basic browsing interactions described in the previous paragraphs. Devices, input modalities and media representations vary in the given examples.

Mobile Video Browsing with a 3D Filmstrip

Mobile devices with touchscreen offer different possibilities of navigation. Swiping and touching are some of the most basic gestures. Multi-touch screens offer additionally the possibility of swiping with more than one finger and zooming/pinching gestures. Hudelist et al. [39] deal with video browsing on mobile devices with touchscreens. They use an approach with a 3D method that uses the metaphor of a filmstrip. Segments of the film are split up equidistantly, with each segment represented by one frame. Figure 2.21 shows a screenshot of the application. Swipe and drag gestures allow navigating through the film. By swiping on the area next to the filmstrip the view moves towards or away from the user. If the swipe is executed on the filmstrip, then the filmstrip winds back and forth. Using two fingers for the swipe gesture the angle of the filmstrip will be changed.

Tapping on a frame starts the playback of the according segment. A pinching gesture controls the granularity of the segmentation, increasing or decreasing the sampling rate respectively.

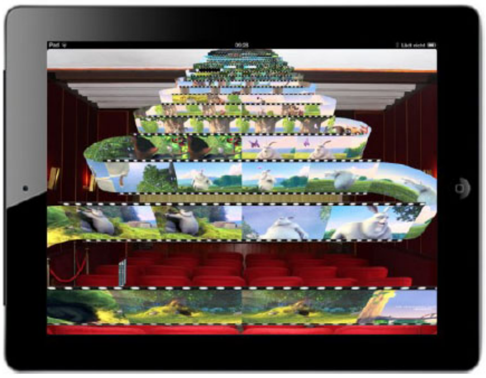


Figure 2.21: 3D Filmstrip after tilting and scrolling [39].

Video Tapestries with Continuous Temporal Zoom

Barnes et al. [10] developed video tapestries for video navigation. The tool combines scene selection provided by a seamless overview of keyframes with the ability to zoom to a more detailed view. The keyframes are arranged on a scrollable timeline and blending between the keyframes is applied to remove the seams that usually exist between them in video browsing timelines (2.22). The authors aim to meet four criterias: *coherence*, *chronology*, *continuity* and *completeness*. Frame boundaries are removed for coherence reasons, because they do not exist in the source video. The tapestry is build from left to right, corresponding to the reading order to meet the chronology requirement. Continuous scaling is demanded for smooth transitions in temporal scaling. The fourth and last requirement is realized by selecting unique visual content for the tapestry, which consists of less pixels than the source video and is therefore not capable to depict the video at a whole.

Video Summagator: An Interface for Video Summarization and Navigation

Video Summagator [69] is a tool for video summarization and navigation. It maps spatial and temporal information into a 3D cube. Figure 2.23a shows the cube for a short video, in which a basketball player moves towards the basket. 3D cubes are composed of voxels, which are defined as being either moving or static depending on motion analysis. There are three parameters that can be controlled by a user: the shape of the cube, the opacity and the sampling rate. The 3D cube can be rotated, translated, scaled and deformed by the user. Shear deformation can be applied on the video cube, either globally or locally. Global shear transforms the cube to a panoramic summarization. Skeleton-based shearing is a local transformation, which arranges video frames along a spline, whereby control points of the spline can be defined by a user. The video frames are visualized

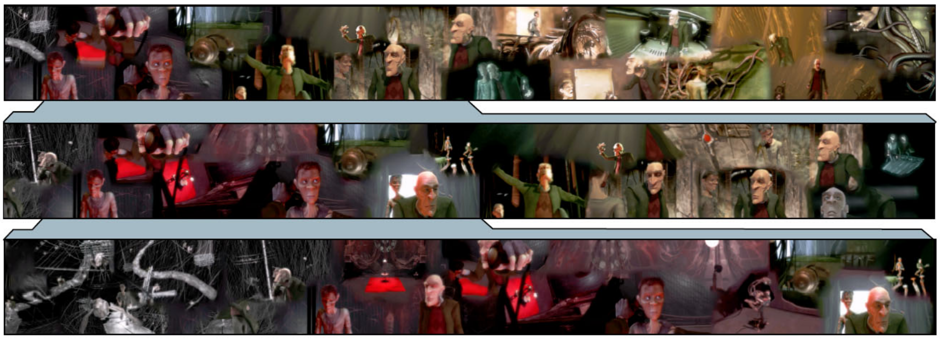
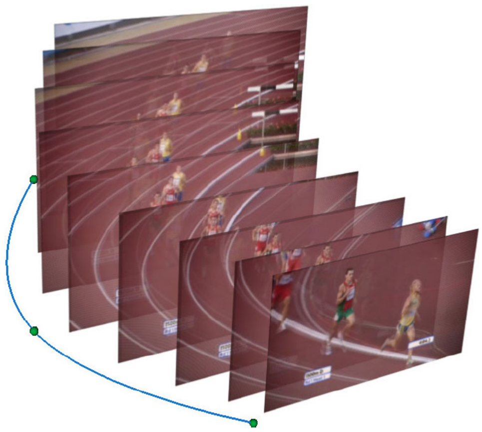


Figure 2.22: Video tapestries [10].

perpendicular to the spline, which corresponds to the time axis. The opacity can be controlled individually for dynamic, static and boundary voxels. The sampling rate can be uniform or variable over time. Skeleton-based shearing is shown in Figure 2.23b. The user can define points along the time axis and move them for shearing. The line on which those points are is parallel to the time per default. Figure 2.24a and 2.24b depict the comparison between different rotations of a video taken with an almost stable camera. The tool is also capable of interacting with panning camera motions, which create a panorama, and moving cameras, which follow actions and video navigation.

(a) Video summary of a basketball video in a 3D cube.



(b) Video summary of a shear view.

Figure 2.23: 3D visualizations of a video with video summagator [69].

DRAGON: A Direct Manipulation Interface for Frame-Accurate In-Scene Video Navigation

Karrer et al. [52] developed DRAGON, an interface for video navigation. It is based on direct manipulation of objects within the video. The name, DRAGON, is short for DRAGgable Object Navigation. A common method of navigation through the dimension

(a) Front view of a video.

(b) Side view of a video.

Figure 2.24: 3D visualizations of videos with video summagator [69].

of time in videos are scrollbars, with a cursor that can be dragged along the timeline to alter the current point in time to show the frame of the video. In this prototype, the user can drag an object in the video along its movement trajectory to move forwards or backwards in time (Figure 2.25). In comparison to common timeline-sliders, where the time is mapped linearly, the figure depicts also how the object trajectory moves against time, which is demonstrated by the varying pace of the object. Karrer et al. [53] furthermore developed PocketDRAGON, a version of DRAGON for mobile devices. The features are almost the same, although the computation is outsourced to a server because of the lower computing power of mobile phones at that time. Karrer et al. mention the drawbacks of the small screen size and therefore the limited space for input elements and the occlusion of the finger when interacting with devices with touchscreens. Furthermore, the control is less accurate than on a desktop device. Touching the device with one finger is associated with an object in the video which can be moved along the trajectory, analogously to DRAGON. Swiping horizontally with two fingers leads to navigation between scenes.

Visualization of Personal History for Video Navigation

Al-Hajri et al. [4] developed the Video History System (VHS). VHS collects information about the user's browsing behaviour and utilizes this history to ease the navigation for them. The prototype consists of three views: the *video library*, the *viewer* and the *history*. The viewer is the element that enables the user to watch videos and the element that records user actions (seek, play, pause, change video,...). The view element furthermore contains a timeline and a filmstrip with thumbnails of video segments, which enables navigation within the video. Therefore the system can track which segments of a video the user is watching one or multiple times and which segments the user skips. VHS has two different views available for the history view: a timeline view and a video tiles view. The timeline view, shown in Figure 2.26a contains a vertical timeline in the middle of the screen, with keyframes arranged on both sides. The elements along the timeline have an indicator to the time the user watched the video. This can be changed by sorting by start time of a video or the most watched videos. The video tiles view (Figure 2.26b) exploits

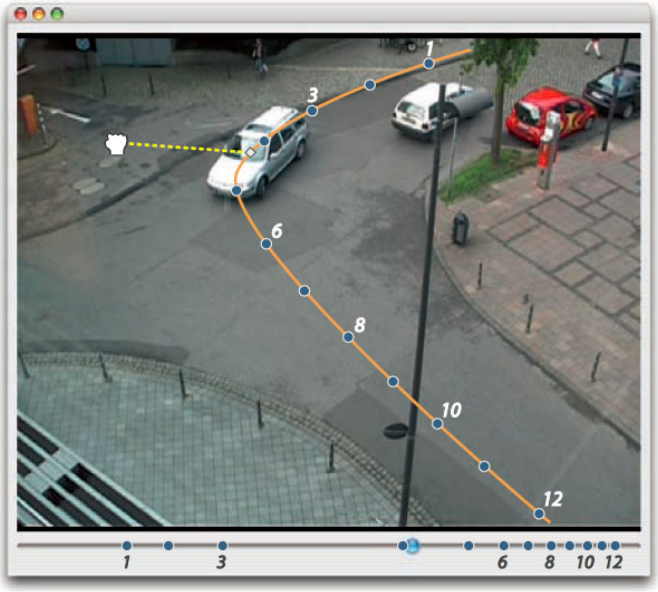
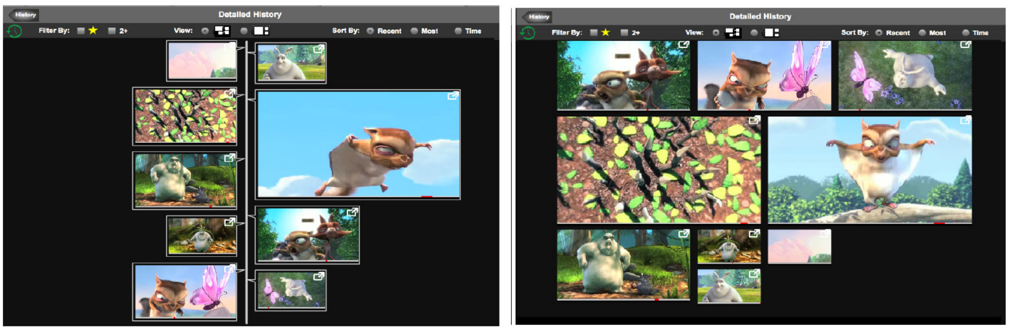


Figure 2.25: DRAGON [52].

the video screen space and still provides an implicit ordering. The size of the keyframes in both views depends on how often a video segment was watched. The keyframes can be small, medium or large; the more often they were watched, the bigger the frame. Furthermore, both views allow the user to filter the video segments by favorites or by the number of views. The visualizations have a limit on how many keyframes they can show in the history at once. If the limit is exceeded the affected frames are stacked. The user is able to zoom into the stack using the mouse wheel, which shows the history with those frames.



(a) History timeline browsing. (b) History tile browsing.

Figure 2.26: Personal history video browsing [4].

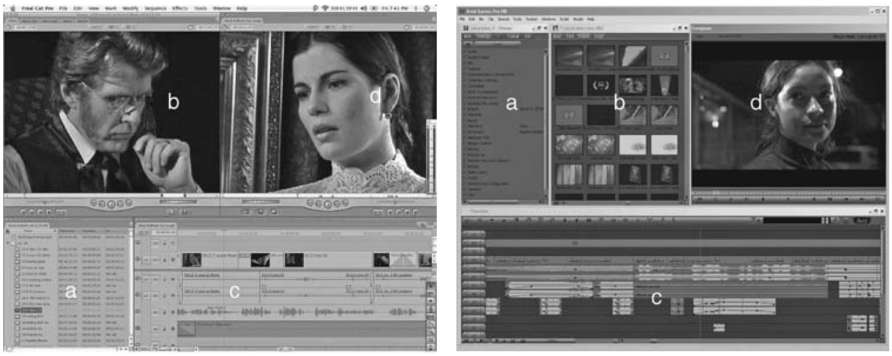
2.4.3 Video Editing

Video editing is a post-processing step in filmmaking. Compesi and Gomez [19] state *"From an aesthetic standpoint editing is the process of arranging individual shots or sequences into an order that is appropriate for the program being produced"* [19]. This corresponds with the aim of this work at hand and is why video editing is an important topic addressed here. Early on in film history, editing was a linear process, a consequence of the physical medium the video was recorded on, the videotape. Non-linear Editing (NLE) is the type of video editing, which emerged with the evolution of digital videos. In contrast to videotapes, digital media allows random access to arbitrary positions within files. Furthermore, Compesi and Gomez [19] mention *insert edits* and *overlay edits*, which are the kinds of editing that became possible through digitalization. Insert editing is adding a shot or sequence between other shots or sequences or in front of them, which shifts subsequent shots behind the inserted shot on the timeline. Overlay editing refers to adding footage on top of other footage. This corresponds to *insert edit* in linear editing, in which it was not possible to add sequences between other sequences. Overlay editing does not extend the duration of a video.

Hurbis-Cherrier [40] sums up four basic windows in NLE softwares: a.) the *Browser*, which contains and represents the media files, b.) the *Viewer window*, whose purpose is to show a preview, c.) the *Timeline*, the place where the editing and arranging of media files takes place and d.) the *Canvas*, which enables the editor to move through media files. Those windows are shown in Figure 2.27 for Final Cut Pro HD and Avid Xpress Pro HD and marked by letters. Ohanian and Philips [70] refer to just three parts of a digital non-linear editing (DNLE) system: *"[...] a footage display area, an editing area, and a graphical view of the sequence. The footage display area is often referred to as a bin, serving as an analogy to the film editor's canvas bin. While the terminology may differ, there are three common concepts to be found in DNLE systems. They are: the clip, the transition, and the timeline"* [70]. Both referenced descriptions of NLEs have two components in common: the footage display area and the timeline. Those incorporate the most important aspects in the work at hand, which aims to develop a visualization and interaction of rearranging images in an efficient way, where time and chronology play an important role. Timelines commonly support multiple tracks for video, graphics and audio. Compesi and Gomez [19] state that user interfaces of editing software usually include a timeline, clips and bins (audio and video material), preview and editing window.

Figure 2.28 shows two screenshots from Adobe's Premiere Pro[6], which is also a software for NLE. The screenshots are taken from a video tutorial, which introduces the user interface. The post was from September, 7th 2018 and is therefore more recent than the screenshots in 2.27. The software offers different predefined workspaces for different purposes. The picture on the right 2.28b for example is for audio. The other workspaces are: Assembly, Editing, Color, Effects, Libraries and Graphics. The user interfaces can be adapted by the user. The picture on the left 2.28a shows the workspace for video

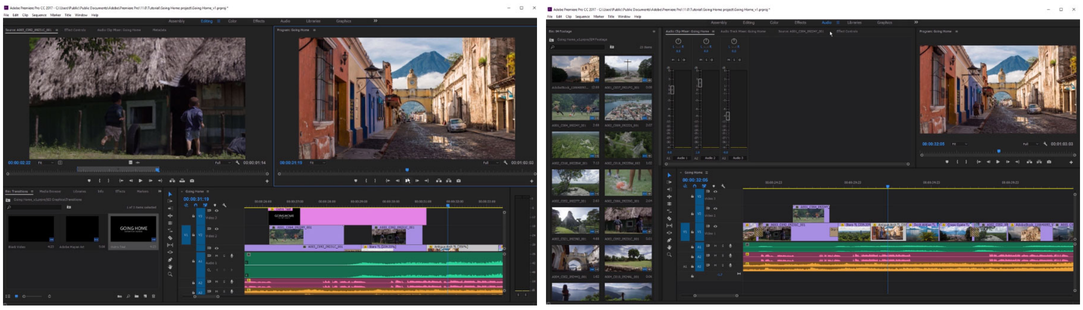
[6] `https://www.adobe.com/at/products/premiere.html`, last accessed: 03/04/2018



(a) Screenshot of Final Cut Pro HD. (b) Screenshot of Avid XPress Pro HD.

Figure 2.27: Examples for presenting the four basic windows in NLE software [40].

editing. It consists of four big areas: the *media browser* in the lower left corner, to browse and access media files, the *timeline* in the lower right corner, the *source monitor* in the upper left corner, which enables the view of clips and adjustments for their use, and the *program monitor*, which shows the result of the created timeline. These areas match the areas, which were mentioned by Hurbis-Cherrier.



(a) Editing workspace. (b) Audio workspace.

Figure 2.28: Screenshots of Adobe's Premiere Pro CC from a video tutorial [47].

In the following various examples are shown that do not follow the common approach described previously. The first method is for pen-based technology, followed by a tool for collaborative video editing on a tabletop surface. The last example builds on an experimental personal user history of viewed clips.

Video Editing with Pen-Based Technology

Cabral and Correia [17] developed an interface for pen-based video editing on tablets. The interface of the prototype is vertically split into two parts (Figure 2.29). A selection of tools and the preview screen is placed at the left part of the interface. Tools offered are, among others, painting, selection, moving and erasing. Available video assets are also located in the left part of the screen. The videos are displayed in one frame, arrows to the left and the right beneath the frame allow a user to navigate through the adjacent frames. A canvas is placed on the right side and embodies a two-dimensional timeline. The tools are used to arrange single frames (frame mode) or a series of frames (segment mode) within the matrix to create a new video. The video is created by composing all frames in the canvas in an order from the top left to the bottom right.

Figure 2.29: Interface of the video editing prototype that makes use of a pen [17].

Designing Environments for Collaborative Video Editing

Terrenghi et al. [86] developed a tabletop application for collaborative video editing. It contains tools that are suited for collaborative editing. Figure 2.30 shows the tabletop and the concept of the surface including the components. The Public Clip-Container holds the video and audio assets and is easily accessible for every member of the group that interacts with the tabletop. The container can be dragged around by grabbing the border with a finger and moving the finger along the screen. Media clips are added by dragging them out of the public clip-container with a touch gesture. A digital wheel is located in the middle which can be used to browse through the material. The shared monitor enables the preview of clips or sequences. Media clips can be sliced and put together, as can be seen in Figure 2.30. Figure 2.31a depicts the control for a video clip. The speed can be changed by stretching or shrinking the element when one touches both ends and moves one's fingers apart or towards each other. Play, pause and scrubbing is facilitated by interacting with the handle of the clip. The connectors merge video sequences. Clips can be cut by pulling together both handle bars, similar to a scissors. The implementation of the collaboration tabletop also includes a hybrid tool, which has the shape of a cylinder with one marker on each end, which determine the current function of the tool. One function is to change the volume, another is to trim the clip after rough cutting. The physical tool can be rotated after it is detected by the table to enable a high level of precision. When a marker of the physical tool is detected, the digital surface displays different elements, which can be seen in Figure 2.31b.

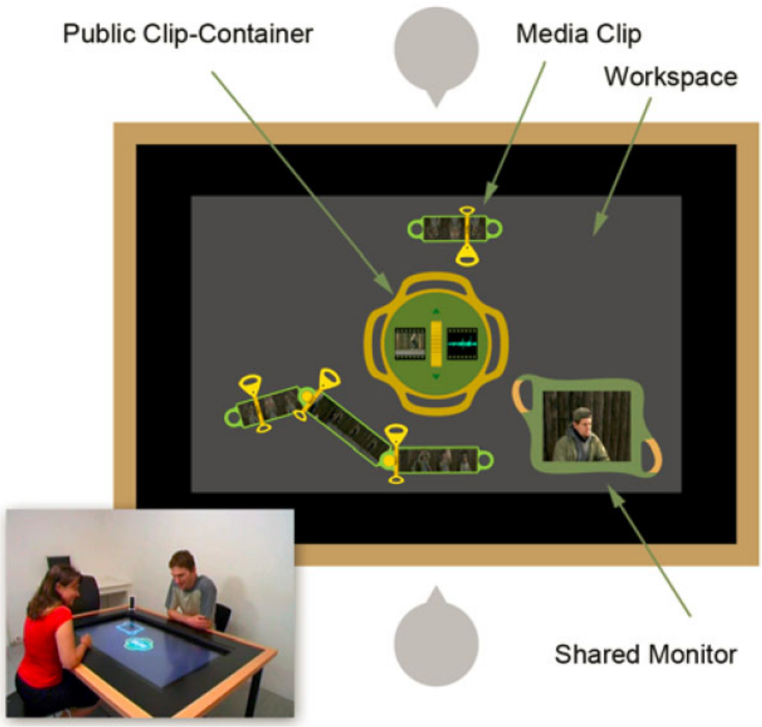
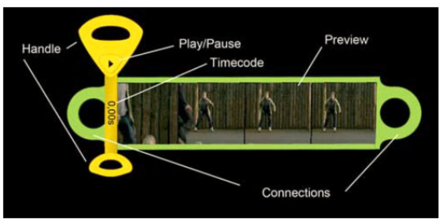
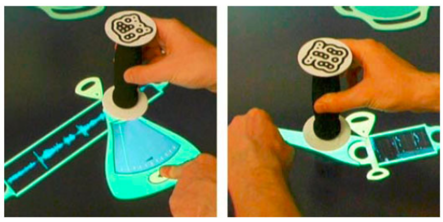


Figure 2.30: Tabletop for collaborative video editing [86].



(a) Clipping tool for video editing on a tabletop.



(b) Hybridtool for video editing on a tabletop.

Figure 2.31: Tools for a tabletop for collaborative video editing [86].

Casual Authoring Using a Video Navigation History

The authors of *Visualization of Personal History for Video Navigation* [4] (Section 2.4.2) developed an authoring tool which is based on the video navigation history [28] (Figure 2.32). The video sequences from the history can be used to create a playlist. The vertical timeline on the left in each screenshot depicts the media history of the user, called the user timeline. A red bar below each video indicates the intervals, which were viewed by the user. The center image is the main player. The horizontal timeline in the left screenshot is called the video timeline. The video timeline represents a video with thumbnails of key frames and it allows the user to navigate within the video. The video timeline can be replaced by the playlist, which is shown at the bottom of the right screenshot. The playlist contains video intervals from the video timeline or the history, which were manually added by the user. The playlist can be saved or exported to a video.

Figure 2.32: History visualization for casual video authoring [28].

2.4.4 Hypervideo and Interactive Multimedia Authoring

To further define the scope of this work, a short detour into interactive multimedia authoring and hypervideos shall be taken. Hypervideos are videos which include hyperlinks to other assets with additional information [35]. There are two major types of hypervideo [64]. The first is *heterogeneous hypervideo*, where hyperlinks refer to texts, images, audio, animation or videos, which provide further information to the current scene. The second is *homogeneous hypervideo*, which is characterized by hyperlinks which are connected to other scenes in the same video, enabling nonlinear navigation in videos. Mixed forms of hypervideo also exist. Hoffmann and Herczeg address hypervideos in combination with storytelling and possibilities for a viewer to interact and to make choices within the story at specific points in time [35]. Yet homogeneous hypervideo is also applicable in areas besides storytelling, with Meixner [64], for example, who mentions virtual tours through houses or cities or e-learning. Figure 2.33 depicts different types of hypervideos and multimedia presentations and their characteristics. Meixner's work covers hypervideos as well as interactive multimedia, which are both a subcategory of hypermedia and therefore consist of hyperlinks between media files. Meixner extracts a few key points from literature, namely that interactive multimedia presentations contain static or continuous media and the existence of temporal and spatial relationships.
Multimedia presentations can be divided into static and dynamic presentations. This distinction impacts user interaction. Static presentations are rendered in advance and offer only VCR controls (play, pause, fast forward, fast backward). Dynamic presentations are composed at runtime, enabling more possibilities for the user to alter the course of the video at runtime.

Type	Media	Interactivity	Temporal synchronization	Spatial layout
Homogeneous hypervideo	video	navigate in graph	no	no
Heterogeneous hypervideo	linear video and media	jump to/interact with media	between main video and media	yes
Hybrid hypervideo	videos and media	navigate in graph and jump to/interact with media	between main video and media	yes
Static multimedia presentation	media	VCR actions	yes	yes
Dynamic / interactive multimedia presentation	media	VCR actions and content selection	yes	yes

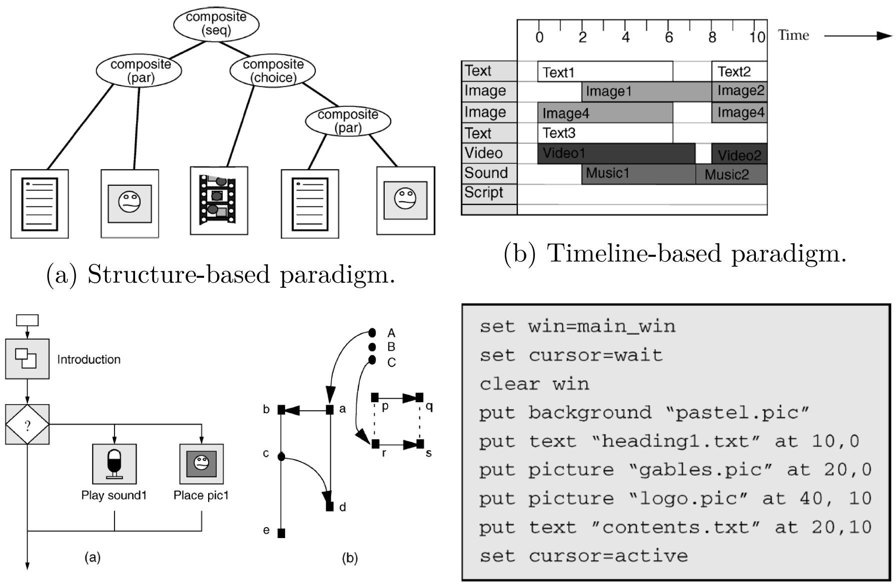
Figure 2.33: Figure shows a table with a comparison of hypervideo types and multimedia presentations taken from Meixner [64, p. 9:6].

Bulterman and Hardman [15] describe a multimedia *authoring system* as follows: *"An authoring system allows the presentation creator to develop a narrative structure based on a collection of media assets and a creative intent that manages the presentation's visual and temporal flow"* [15, p. 90]. Their specification covers the use of NLEs and the importance of the temporal dimension gets emphasized, but they highlight one difference to multimedia presentation, the lack of interactivity in films. However, there are enough commonalities, so that this work may benefit from the insights from this area. Bulterman and Hardman report on four paradigms for authoring: *structure-based*, *timeline-based*, *graph-based* and *script-based* (Figure 2.34). Structure-based authoring shows the types of media files and their relations. The structure-based type contains information about media assets and type of composition, which relates to the ordering and thereby the activation time. Timeline-based authoring visualizes media files stacked separately in one line, with the starting time and duration of each item indicated by a bar in the according line. Graph-based authoring represents media assets and relations as nodes, which are connected with edges to indicate which assets are affected by which of the actions. Script-based authoring provides a window to write and edit scripts.

Hardman et al. [33] address the time aspect in hypermedia. They present four subtypes of presentation time: *media element time*, *document time*, *rendered time* and *runtime*. Each type is connected to a different stage within the authoring process of elements. They are listed below:

- *Media element time* is the duration of a media asset.
- *Document time* corresponds to the duration of the media in the document that is determined by the author.
- *Rendered time* is the actual time that is needed to be played-back. This is important in cases where media assets have alternatives depending on system requirements or interactivity. If no alternatives exist, rendered time matches the document time.
- *Runtime* is the duration of the presentation, which again can vary if alternatives or interactivity is available.

In media authoring, representation and editing are the cornerstones to create multimedia content that consists of assets. Duration, transition and looping effects may be assigned by the author to media elements. The authors of the paper state the requirement of defining temporal links and relationships if multiple media elements are used. Rendered time will be neglected in this work, because interactivity based on user choices will not be incorporated.



(a) Structure-based paradigm.

(b) Timeline-based paradigm.

(c) Graph-based paradigm.

(d) Script-based paradigm.

Figure 2.34: Depictions of multimedia authoring paradigms [15].

Hardman et al. [32] dissect media production into canonical processes, as an initial step to improving interoperability of multimedia systems. They have extracted nine steps, whereby two of them, *premediate* and *create media asset* are not of interest for the purpose of this work, since the focus here lies on editing. The remaining seven, *annotate*, *package*, *query*, *construct message*, *organise*, *publish* and *distribute* will be further investigated for their relevance of editing.

Baecker et al. [8] developed an authoring tool for dynamic visual presentations like motion pictures called MAD (Movie Authoring and Design). They set four key design goals for their prototype: *Idea structuring*, *Multimedia support*, *Visualization* and *Interchange representation.* For a good authoring tool Becker et al. propose the presence of both, a view of the result and a view of the progress. Furthermore, they recommend different representations for script, storyboard and movie playback.

CHAPTER 3

Methods

This section explains the methods which are applied in this thesis. It describes the steps which were performed to be able to answer the research questions. First, an introduction to *User-Centered Design* (UCD) is given, emphasizing the importance of user involvement in the design and implementation of this work. Subsequent descriptions of each method offer an outline for readers, who are unfamiliar with the topic. In this thesis the methods *Interviews*, *Focus Groups*, *Prototypes* and *Usability Testing* were applied and are therefore covered in this section. A literature review was conducted as a first step of the thesis and is covered in depth in Chapter 2.

3.1 User-Centered Design

The approach in this thesis follows the user-centered design principle. UCD is part of interaction design, whereby interaction design refers to the design of artifacts people interact with [75]. Sharp et al. state that the difference of Interaction Design (ID) and Human-Computer Interaction (HCI) is the scope. According to the authors, HCI is concerned with computer systems, in contrast to ID, which is multidisciplinary.

UCD has its origins in participatory design, which evolved from a labor movement in Sweden and Norway. Social factors gained importance in the 1970s at people's workspaces which involve technologies. The attention at workspaces changed from the technology to the user and how to integrate them in the design process [84][68].

The aim of user-centered design is to include human factors in the development of a system. Potential users are incorporated to enable decision making that is founded on conducted research. UCD methods are applied to investigate requirements and to maximize factors of usability [23][84].

ISO 9241-210:2010 defines usablitity as *“extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use”* [29].

The term *User Experience* (UX) is often used in combination with design and goes beyond usability. Although UX is related to usability, they are not the same. UX is harder to grasp and authors have various definitions for UX [14]. ISO 9241-210:2010 describes user experience as: *“person’s perceptions and responses resulting from the use and/or anticipated use of a product, system or service”* [29], followed by three notes. The first note says: *“User experience includes all the users’ emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviours and accomplishments that occur before, during and after use”* [29]. User Experience goes beyond usability. The last note in the standard adds that *“Usability criteria can be used to assess aspects of user experience”* [29].

Gould and Lewis state three principles interaction design rests on, which are: 1.) trying to understand the users and their context early, 2.) get feedback of users to scenarios, prototypes and similar throughout the development and 3.) alter and refine the design iteratively by conducting user tests and applying their results. Those three principles are called “Early Focus on Users and Tasks”, “Empirical Measurement” and “Iterative Design” by Gould and Lewis [31]. Deuff and Cosquer [23] refer to ISO 9241-210 from 2010 in their book to describe six principles of UCD:

1. *“The design is based upon an explicit understanding of users, tasks and environments.*
2. *Users are involved throughout design and development.*
3. *The design is driven and refined by user-centered evaluation.*
4. *The process is iterative.*
5. *The design addresses the whole User eXperience.*
6. *The design team includes multidisciplinary skills and perspectives.”* [23, p. 15]

Figure 3.1 depicts the ISO standard in four main stages. The ISO standard defines human-centered design as *“approach to systems design and development and development that aims to make interactive systems more usable by focusing on the use of the system and applying human factors/ergonomics and usability knowledge and techniques”* [29]. Furthermore the standard makes a declaration for the difference between “human-centered” and “user-centered”. User-centered focuses on the user of the system, human-centered takes all stakeholders into account.

UCD is an iterative process. Figure 3.2 shows a diagram of an interaction design lifecycle from referred literature. Various methods exist to get an understanding of the user,

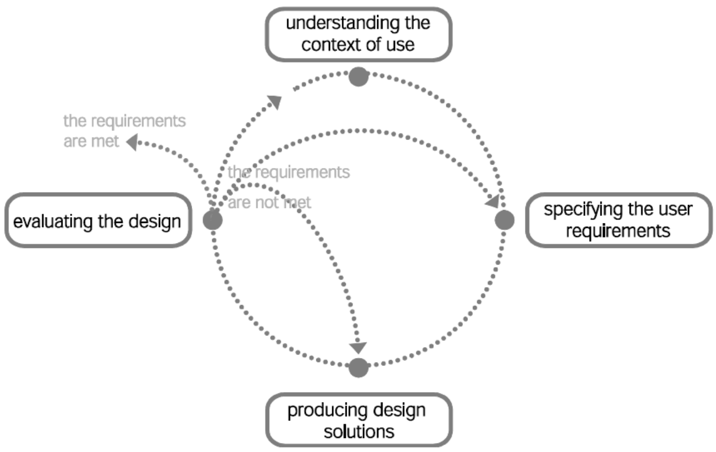


Figure 3.1: Graph of the phases of users-centered design according to the ISO standard [23].

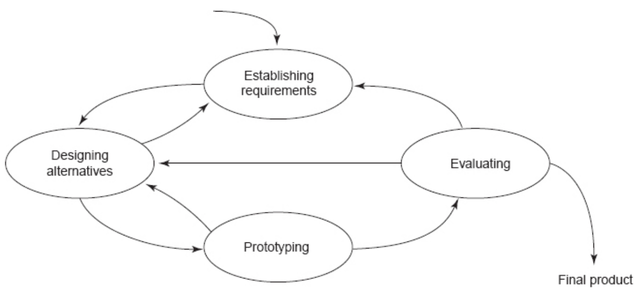


Figure 3.2: Interaction design lifecycle [75].

these also vary in the point of time and the step within the design process. Kuhn and Muller published a diagram for user involvement over time in the development process (Figure 3.3). There is also a classification of levels of participation in the design process: informative, consultative, participative and designer, in which the user is the designer of the system [14]. Furthermore Deuff and Cosquer write about techniques within UCD and emphasize four important techniques in agile software developing: user analysis, prototypes, usability evaluation and iterative design.

3.2 Interviews

Interviews are conversations of two or more persons, in which one person (interviewer) seeks information from another person (interviewee). Interviews can be done with multiple interviewees at a time [12][74]. Interviews are classified as one of four basic types: unstructured, structured, semi-structured and group interviews. The first three types relate to how much control an interviewer has over a conversation. The last

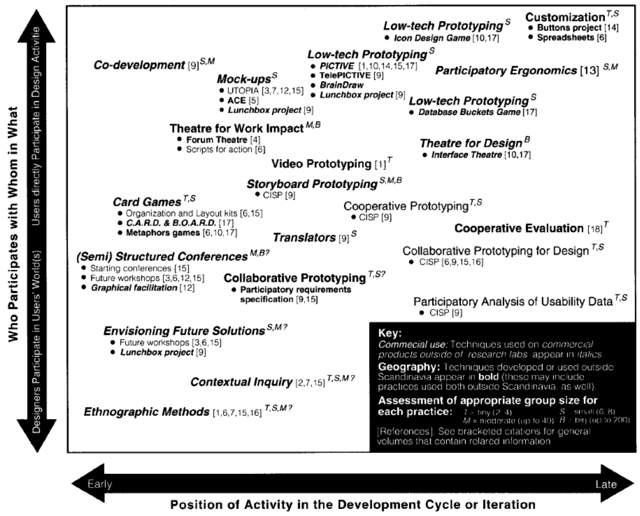


Figure 3.3: Graph of methods in participatory design [68].

type is discussed in Section 3.3 (Focus Groups). Interviews in general are a qualitative research method, however, interviews can be structured very strictly and the resulting data becomes more quantitative in such cases. Structured interviews can be compared to questionnaires, which are usually conducted online or on paper. Questionnaires consist of a set of questions that have to be answered by the participants. They aim to collect results from many participants and are therefore rather quantitative than qualitative [1]. Interviews can be conducted in person or be technology-mediated. The decision for the type of interview and the design depends on factors like purpose, research question and the point in time within the study the interview is conducted.

Structured. This type of interview offers the most control for the interviewer. Structured interviews are similar to questionnaires. The interviewer sticks to prepared questions. A major part or all of the questions may be closed-end with a predefined set of possible answers. Data from the interviews are comparable because of the predefined guideline, therefore the interviews are standardized.

Unstructured. Unstructured interviews offer freedom to the interviewer and the interviewee. Questions are open-ended and the interviewee may answer questions in a preferred scope and detail. The answers do not have to follow a specific format or fulfill expectations and may reveal information that was not considered before by the researcher. The interviewer can follow the interview guide or improvise on the spot to follow an interesting topic, which came up unexpectedly in the course of the interview. In contrast to structured interviews it is harder to analyze the qualitative data from the interviews.

Semi-structured. Semi-structured interviews are a mix of structured and unstructured interviews. They may contain open-ended and closed-ended questions. The interviewer may deviate from the interview guide, however, a basic set of questions for key points should be covered in every interview.

In this thesis, interviews with video editing experts will be conducted. The interviews will be build up on the results of the focus group. The interview will be split into a theoretical and a practical part that is corned with a mockup.

3.3 Focus Group

Group interviews, also called *focus groups* are interviews with more than one person at a time [27][12][1][75]. Apart from the group size, focus groups are similar to interviews. Focus groups are group discussions, in which opinions and experiences about a topic are exchanged from which researchers can benefit. Statements about the size of focus groups vary in literature. Three to ten people, three up to six or seven, and six to ten participants are stated in different sources. Focus groups consist of a number of participants which represent the target population. The focus group can profit from group dynamics and input from one participant may stimulate other participants. The setup of a focus group aims to reach a close to everyday life and supportive environment. The facilitator guides the focus group, he or she might be flexible to deviate from the priorly created guideline and to encourage every participant to express his or her opinion. A task may be prepared by the facilitator to be elaborated by the participants. Focus groups are a suitable method in the ideation phase. Common durations of focus groups are one to two hours.

In this thesis the method is applied for ideation and requirement analysis. It builds up on the previous conducted review of suggestions from about 140 students. The focus group will first individually and later collaboratively elaborate and discuss a prepared task.

3.4 Prototypes

Prototypes are a method in the design process that can be used to embody the current concept or parts of the current concept. Outcomes from previous methods in the process are structured and evaluated in order to build the prototype [23][50]. Saffer describes the prototype as a *"... tool for communicating. Prototypes communicate the message 'This is what it could be like' "* [79]. One or more prototypes can be crafted throughout the design process. Because prototypes represent a product or parts of the product in intermediated steps of the process, they can be used for evaluations with users to reveal weaknesses and strengths in the current design. Although Holmquist states a difference between mockups and prototypes in their functionality, this difference will not be made in this thesis. Holmquist defines mockups as tools, which have only the appearance but no functionality, in contrast to prototypes, which are functional but have a lack of appearance [36]. Houde and Hill state three dimensions of prototypes: look and feel, role and implementation. The first dimension refers to sensory aspects of the prototype, that are felt by the user. The second dimension describes the function a prototype offers. The last dimension, implementation, describes the prototype itself and how it works [37]. The scope of prototypes range from low-fidelity prototypes to high-fidelity prototypes. Both ends of the continuum are described in the subsequent sections. The type of prototype that should be used depends on the product that is designed, the available resources and the purpose of the prototype. The intended aim or the question a researcher wants to address with the prototype itself or the testing of the prototype are important for the design of the prototype.

3.4.1 Low-Fidelity Prototypes

Low-Fidelity (LoFi) prototypes are usually simple and fast to build and made out of cheap material. They are suited for early phases of the design process and facilitate the exploration of the product [75]. Paper is often mentioned as material for the creation of LoFi prototypes, although they can also be implemented digitally. Storyboards are one example for LoFi prototypes. Storyboards are a series of sketches that depict a scenario in which the system is used. LoFi prototypes can also be embodied as physical prototypes, which can be composed of clay or wood or other materials. LoFi prototypes usually do not actually work or work only to a certain extent, but they can be "brought to life" for testing reasons by the person who conducts the test. This is called "Wizard of Oz", because someone behind the curtain only simulates the effects of a user action. The name refers to the story by Baum and Denslow from 1900.

In this thesis a LoFi prototype is used as a basis in the practical part of the interviews. The mockup is implemented digitally and consists of single screenshots. The screenshots are presented on a smartphone and do not have a functionality except switching to the next or to the previous screenshot. The screenshots are used to present the intended functionality, before the functions are actually implemented.

3.4.2 High-Fidelity Prototypes

High-Fidelity (HiFi) prototypes are more sophisticated in creation and functionality. HiFi prototypes are supposed to act like the final product from the sight of the user, but they do not have to actually work like the final product. Complicated functionalities may be faked. Important is the image the user perceives in terms of functionality and aesthetics [79]. Parts of the prototype might be reused in the final product. HiFi prototypes need more time to create and are therefore more expensive. Thus high-fidelity prototypes usually find their usage later in the design process.

In this thesis a HiFi prototype will be implemented to test the outcomes up to that point. The key aspects which appeared in the process of the thesis will be implemented in a digital prototype. In contrast to the LoFi prototype, the HiFi prototype will have functionality, therefore the theory from the previous work will be put into practice. Transforming theoretical findings into a practical prototype makes it possible to test those findings.

3.5 Usability Testing

Usability tests are conducted to detect possibilities for improvement and to find weak spots. Preece et al. describe usability testing as a quantitative method which draws on satisfaction questionnaires and measurable factors such as time needed to complete a task, number of errors during execution, number of keystrokes, etc. However, usability testing can also be approached qualitatively, like considered by Lazar et al. [57] applying methods like observation or asking participants to say their thoughts out loud ("thinking aloud"). Observations as a method is borrowed from the scientific discipline ethnography. While ethnographers use observations to better understand people and the context people live and work in, in usability testing observations aim to detect possible weak spots in a proposed design. Usability testing can be conducted in a controlled environment (lab studies) or in a natural environment (field studies) [75]. Formative usability tests are conducted early in the development process with low-fidelity prototypes. The goal of formative tests is rather gathering exploratory insights. Summative tests on the other side are operated with high-level prototypes, aiming to collect information about effectiveness of the developed design. Formative tests might rather yield qualitative data while summative tests might rather yield quantitative data.

In this thesis usability testing is used at the end of each prototyping iteration. A mockup will be discussed with experts and the functional prototype will be tested with a number of participants similar to a lab study. Usability testing in this thesis is always conducted in a qualitative way, applying methods such as observations, thinking aloud, and interviews.

CHAPTER 4

Design and Development Process

This chapter describes the implementation of the thesis. Each method used, which was applied and how it was applied, is described in detail. The approach and setup of each method will be declared, before outcomes and influences on the work will be discussed. Before conducting the focus group, student assignments of a specific task were analyzed. Assignment analysis was the first step after the literature review and corresponds to the first section in the chapter. The focus group was based on the results of the assignment analysis. Items and categories were deduced from the student concepts, which were presented in the focus group as building blocks. A mockup was developed based on the outcomes from the focus group. The process of the mockup design is described in Section 4.3. Interviews were conducted in a final step, before the implementation of the prototype started. The implementation was split into three milestones. The first two milestones were finished with user evaluations at the end. The final prototype was tested more extensively. The prototype as well as the user evaluations are discussed at the end of this chapter.

4.1 Review of Suggestions

A review of suggestions was part of the thesis. A similar assignment was part of a course from the bachelor program of media informatics. The course is called "Interface and Interaction Design" (IID). The specification of the assignment was as follows: a mobile phone with touch-input is given. There are between 100 and 300 images from the last holiday, which are not in the favored order.The students were asked to design a touch-based interface for a mobile phone, which enables an easy rearrangement of 100 - 300 images in a manual way. For example, to change the sequence from 1-2-3-4-...-300 to 101-1-2-234-236-...-15. Metadata can be used for supporting the task, but are optional. A hint is given, that says that the interface can support the rearrangement of single images as well as group of images. The original specification of the assignment is written in german as follows:

"Sie haben ein Mobiltelefon mit Touchinteraktion (zB. Nokia Lumia 520).
Auf dem Mobiltelefon befinden sich 100 bis 300 Fotos von Ihrem letzten Urlaub.
Diese Fotos sind nicht in der gewünschten Reihenfolge (zB. ist das schöne Foto vom Hotel ganz am Schluss der Bildergalerie, soll aber zu Beginn gezeigt werden).
Entwerfen und beschreiben Sie ein touchbasiertes Interface für ein Mobiltelefon, mit dem sich die Reihenfolge (der Anzeige) von 100-300 Fotos leichtgängig manuell verändern lässt.
zB:
vorher: 1-2-3-4-5-6-7-8-9 ... 300
nachher: 101,1,2,234,235,236,7,3 ... 15
Metadaten wie Datum und Ort können als Unterstützung verwendet werden, müssen aber nicht verwendet werden.
Überlegen Sie auch, wie Fotos einzeln oder als Gruppe verschoben werden können."

The assignments consisted of text and sketches. Students were encouraged to develop more than one concept and decide for their favorite and describe it in more detail. 141 assignments from 141 students were collected and reviewed. The results of the analysis of the review were taken as input for the focus group, which was the next step of the process.

4.1.1 Analysis and Results

Each assignment was reviewed one by one. The concepts were reduced to items, which were grouped together subsequently to build categories and the categories got descriptive names. The identification of items and categories was iterative. Items are self-explanatory elements. They can be picked from categories to assemble an approach for the task. Assignments which were based on a wrongly interpreted specification were neglected. This was the case with interface designs, which did not include the option for an manual arrangement but only arrangement based on metadata.

The result of the first iteration were twelve categories: “Display”, “Views”, “Interaction”, “Gestures”, “Buttons”, “Mechanisms”, “Selection Methods”, “Order”, “Principle”, “Marking”, “Arrangement Method” and “Sorting”. In the subsequent iterations the items were abstracted and summarized. Items which meant the same but were named differently by students in assessments were combined to one item in the list of the categories. Some items and categories were unmodified in one iteration, while others were modified. Categories were divided into a hierarchy. The purpose of the final categories and items were kept in mind. The purpose was to process the data in a way that makes it possible to present them in a comprehensible way to the focus group. Therefore priorities were set for categories and items, to not exceed the duration of the focus group and the duration of the presentation held at the beginning of the focus group. After the last iteration there were eight categories in three superior categories. The categories and their items are listed in Table 4.1.

Terms

This subsection describes the categories in a hierarchical order, which were extracted from the assignments.

Display. Display describes visual characteristics of the interface, that are considered important for the purpose of the topic. Here, it consists of two subcategories: representation and views.

Representation. The term representation was chosen to describe the way, in which a collection of images is displayed on the screen. The most common representations, which appear in literature for image and video browsing, are grids and timelines. Images are displayed as thumbnails, which are small versions of the images. Schematic sketches for each representation can be found in Figure 4.1.

Grid. All images are arranged as thumbnails in a grid (Fig. 4.1a).

Timeline. Thumbnails are arranged in a sequence. The direction of the timeline can be horizontal or vertical (Fig. 4.1b).

List. Thumbnails are located next to metadata of the corresponding image. Each image is represented by one entry, the entries are arranged one below another (Fig. 4.1c).

Cover Flow #1. Images are displayed frontally in a sequence, the current image is upscaled, pictures close to the current image have a linear scaling factor, images further away are only represented in a small version (Fig. 4.1d).

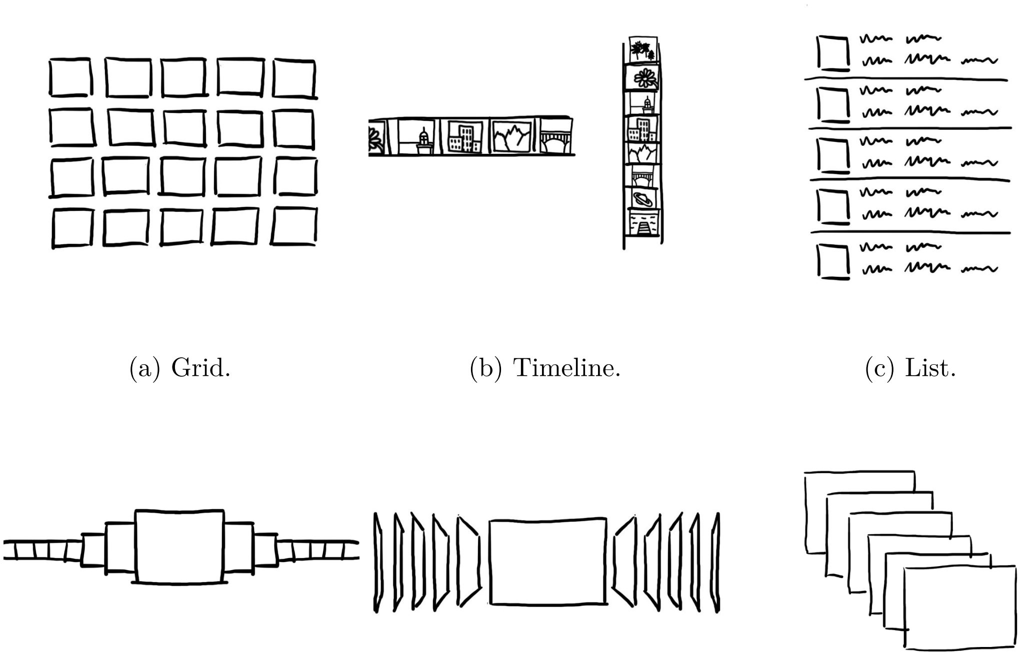
Cover Flow #2. Images are arranged upright, only the current image is shown frontally. Pictures close to the current image are perspectively distorted. The farther away an image is from the current image the closer to 90 degrees the angle of distortion gets (Fig. 4.1e).

Aero Flip. Images are displayed frontally, the active image is in the front, the other images are displaced to the back. The user can scroll through them to change the active image (Fig. 4.1f).

View. A view displays one or more pieces of information, which were categorized as visual characteristic. Views in this sense can be imagined as windows, which are able to show different aspects of the image collection or which provide usable support mechanisms for the user. Those support mechanisms, like the clipboard, add options for the given task. Representations of images are also supposed to be shown within a view. Views can be combined as needed. The size a view takes on the display can be freely determined. Besides views that

Table 4.1: Resulting Categories and Items from the IID Suggestion Analysis

Main Category	Subcategory	Items
Display	Representation	Grid
		Timeline
		List
		Cover Flow #1
		Cover Flow #2
		Aero Flip
	Views	Fullscreen
		Fine / Coarse View
		Clipboard
		3D View (if available)
		Calendar / Date View
		Old / New Order
		Overview Current Selection
		Aggregations
Interaction	Gestures	Gestures & Multitouch Gestures
		Hot Corner
		Hot Area
		Swiping Images
	Selection Method	2 Modes - Single / Multiselection
		Lassotool
		Touch & Hold
		Selection Rectangle
		Swiping
		First & Last Image Selection
	Method	Order of Touch
		Swap
		Position
		Arrange
		Cut & Paste
	Aggregation	Group / Album / Folder
		Tag
		Priority
		Weighting
		Favorites
System	Sort	by Metadata
		by Groups
		by Tags
	Filter	by Priority
		by Weighting
		by Favorites



(a) Grid. (b) Timeline. (c) List.

(d) Cover Flow #1. (e) Cover Flow #2. (f) Aero Flip.

Figure 4.1: Sketches for representation methods of images.

can be shown in parallel on the screen, the interface can be designed in a way in which the user switches between one or more views. Schematic sketches for the views can be found in Figure 4.2.

Fullscreen. The user can watch a single image on fullscreen (Fig. 4.2a).

Fine / Coarse View. The user has a fine and a rough overview of the images (Fig. 4.2b).

Clipboard. The user has a clipboard, where images can be stored temporarily (Fig. 4.2c).

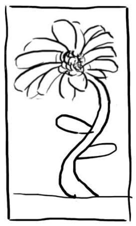
3D View. The user is able to change the view to 3d, if the representation offers this view (Fig. 4.2d).

Calendar / Date View. Images are shown with a chronological context which includes dates (Fig. 4.2e).

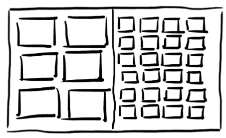
Old / New Order. The user has access not only to the current (new) order of the images, but also to the original order. Either at the same time or by switching between them.

Overview Current Selection. The interface is able to display an overview of the current selection of the user (Fig. 4.2g).

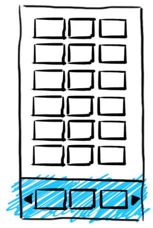
Aggregations. The user has an overview of the created aggregations (Fig. 4.2h).



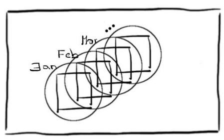
(a) Fullscreen.



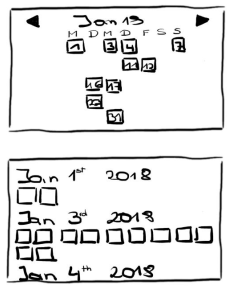
(b) Fine / Coarse View.



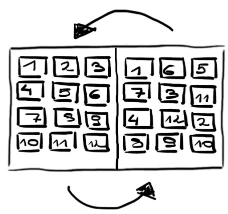
(c) Clipboard.



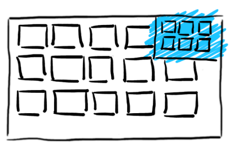
(d) 3D View.



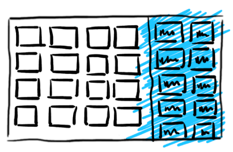
(e) Calender.



(f) Old / New Order.



(g) Overview Current Selection.



(h) Aggregations.

Figure 4.2: Sketches for view methods of images.

Interaction. This category describes the way in which the user can interact with the system to achieve the intended result. The interaction happens via the user interface, which consists of the one or more views and one or more representations of the images. The interaction is split up into four subcategories, which describe principles of the interaction and also for the specific task of rearranging images on mobile devices. The four subcategories are as follows: gestures, selection methods, arrangement methods and aggregations. The last principal, aggregations, is, on purpose, not elaborated in detail, because they also leave a number of options how to implement those principles. They will be discussed if they turn out to be of interest.

Gestures. This category includes single and multitouch gestures. They are introduced as a pool of options to interact with the system, they are decoupled of specific purposes. Therefore it is left open, which gesture is used to achieve something within the application. The first two listed items were already covered in Section 2.2.2. Schematic sketches for the other three listed items can be found in Figure 4.3.

Gestures & Multitouch Gestures. This subcategory includes a collection of gestures. Besides gestures like pinching and swiping the collection consists of gestures taken from Gestureworks[1]. The collection can also be extended by individual gestures. Multitouch gestures were taken from gestureworks.com and are located seperately.

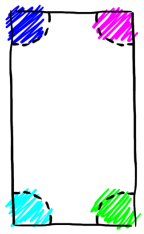
Hot Corners. Hot Corners[2] is a concept, in which the user drops an item in one of the corners. Each corner has a different function, the functions can be freely chosen. For example if the user drags and drops one image in the either lower corner it could be deleted from the collection. Dropping the image in the left upper corner could add the image to a group. The advantage of hot corners is that they save space on the screen, because they are not visible till an event occurs (Fig. 4.3a).

Hot Areas. Hot Areas[3] is a concept, which is similar to Hot Corners. The difference is, that they are not limited to corners but areas, in which the user can drop images. The areas can be positioned anywhere on the screen (Fig. 4.3b).

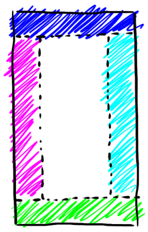
Swiping Images. Swiping images into a specific direction can be coupled with various effects. It is also similar to hot corners and hot areas, with the difference, that images do not have to be dropped in a specific area, but swiped into a direction. The directions can be assigned to specific functions (Fig. 4.3c).

Selection Methods. This topic includes options for picking multiple images at once for further actions. Since the application is supposed to cope with a vast amount of images it is advantageous to pick more than one image at once. The content of this subcategory is a collection of suggestions to achieve this. Selection modes are depicted in Figure 4.4.

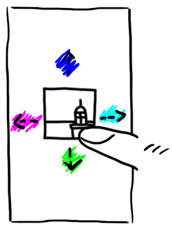
[1] `http://gestureworks.com`, last accessed: 04/26/18
[2] In the focus group guide originally called “Magic Corner”
[3] In the focus group guide originally called “Magic Area”



(a) Hot Corners.



(b) Hot Areas.



(c) Swiping Areas.

Figure 4.3: Sketches for gesture-related interaction.

2 Modes - Single / Multiselection. The user can choose between two selection modes, if the selection mode is on. The single selection mode allows a user to select one image with a single tap. If the multi selection mode is on it is possible to select an arbitrary number of images with multiple taps (Fig. 4.4a).

Lassotool. If a user wants to select images with the lassotool, an arbitrary, closed shape can be drawn. All images that are located within the closed shape are selected subsequently (Fig. 4.4b).

Touch & Hold. Users switch into the selection mode by tapping and holding one image. Users are able to choose further images to add to the selection by tapping on them. Holding each image is not necessary if the selection mode is already activated (Fig. 4.4f).

Selection Rectangle. Images are selected by dragging a rectangle on top of the images. All images within the rectangle are selected, similar to the lassotool but less flexible (Fig. 4.4c).

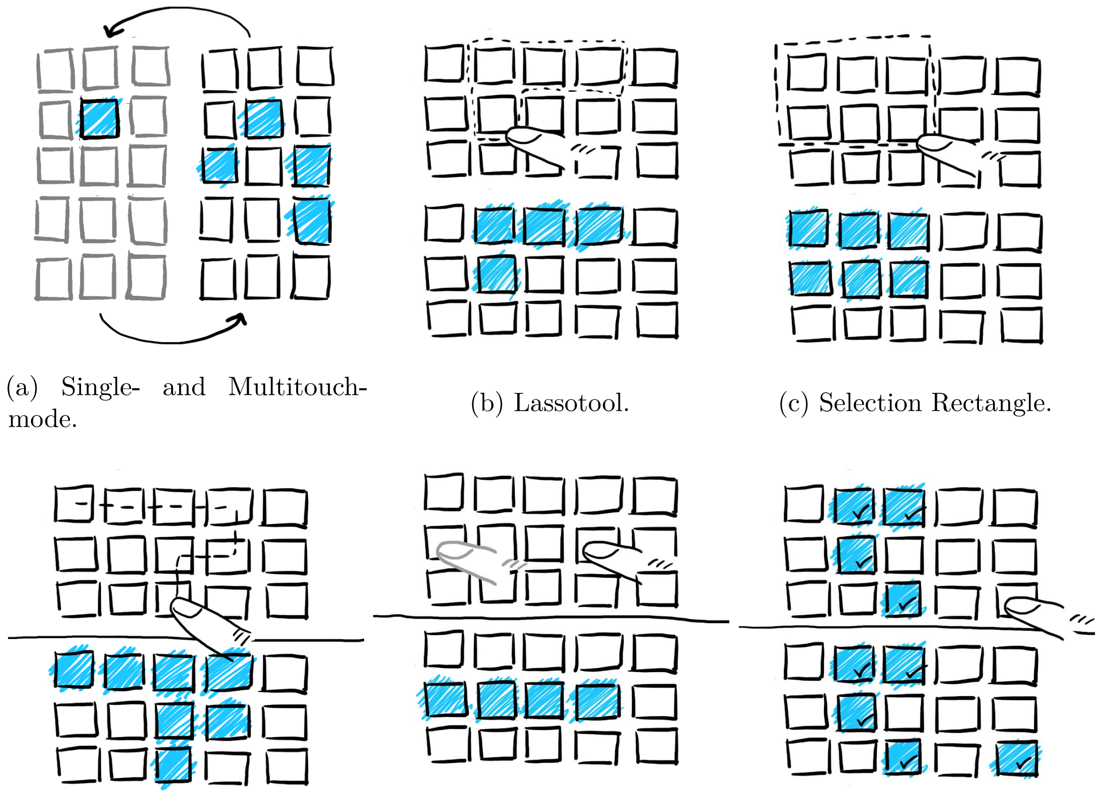
Swiping. Swiping over the screen selects all images that were touched (Fig. 4.4d).

First & Last Image Selection. The user taps on two images, all images between the first and the last tapped images will be selected (Fig. 4.4e).

Arrangement Methods. This category describes approaches for the arrangement and the rearrangement of images. Some methods are inherently coupled to gestures or other interaction aspects, others offer different possibilities for the implementation and can therefore be realized by other techniques. The methods are depicted in Figure 4.5.

Cut & Paste. The new order is created by cutting images and pasting them before, after or between other images (Fig. 4.5a).

Drag & Drop. The new order is created by moving the images in the representation (Fig. 4.5b).



(a) Single- and Multitouch-mode.
(b) Lassotool.
(c) Selection Rectangle.
(d) Line Selection.
(e) First & Last Image Selection.
(f) Touch & Hold.

Figure 4.4: Sketches for selection methods. The upper half consists of the state before the gesture and a gesture depiction, the lower half represents the state after the gesture was performed.

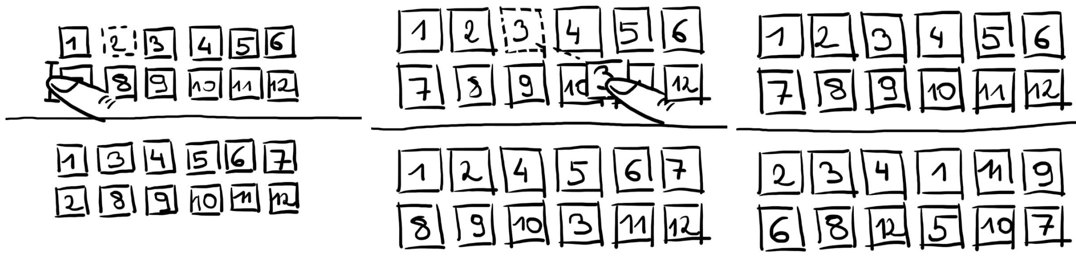
Position. The new order is created by assigning a fixed position number to the images (Fig. 4.5c).

Swap. The new order is created by swapping images with each other (Fig. 4.5d).

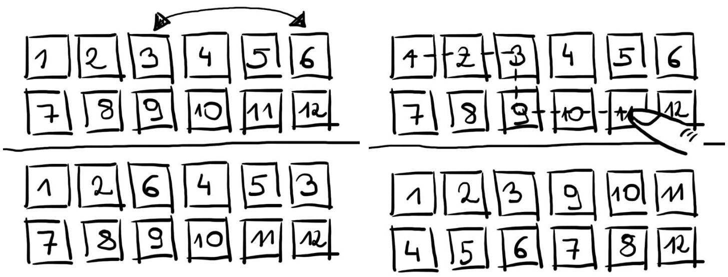
Order of Touch. The new order is created by arranging images according to the order in which images were touched. This can happen by tapping a number of images one after each other or by swiping over images (Fig. 4.5e).

Aggregations. This category introduces a number of options for grouping images. The application should be able to deal with a high amount of images. It is possible that aggregations ease the handling with this amount of images or ease further steps in the process to achieve a specific order of images.

Group / Album / Folder. The user is able to create groups, albums or



(a) Cut & Paste. (b) Drag & Drop. (c) Position.



(d) Swap. (e) Order of Touch.

Figure 4.5: Sketches for arranging methods. Upper half consists of the state before the gesture and a gesture depiction, lower half represents the state after the gesture was performed.

folders. Arranging them in a hierarchical way is possible. Images can be added to those aggregations. The terms may mean the same or differ depending on their implementation.

Tag. The user is able to create tags. Those tags can be used to associate images with keywords.

Priority. The user can assign priorities to images.

Weighting. The user can assign weights to images. In contrast to priority this offers more gradations than priority does.

Favorites. The user can mark images as favorites. In contrast to priority and weighting, which can have a number of gradations, favorites can only be marked as favorite or not.

System. This category includes possible features, which are provided by the system to ease a task. The criteria can be something that was defined by the user or retrieved from the metadata of the image. Sort and filter are supposed to support the user to achieve a task, by a systematic pre-selection or arrangement of images. This could be of help if the user is looking for one or more images when there is a rough idea of the wanted image.

Sort. Sorting arranges images according to one or more criteria.

Filter. Filtering displays only images which fulfill one or more predefined criteria.

Sort & Filter Criterias. The user is able to sort or filter by criteria like:

- metadata
- groups / albums / folders
- tags
- priorities
- weight
- favorites

Two approaches appeared often in the elaborated suggestions of the students. Those two principles are hereinafter referred to as "hold and drag" and "splitscreen". Although they differed in some details and offered options in addition to the basic approach, the underlying principle behind the suggestions was the same. Both principles are described in the Sections 4.1.1 and 4.1.1.

Hold and Drag

The hold and drag approach represents the images as a grid of thumbnails. Thumbnails can be moved by touch and hold on one of the images, and drag it to the new position. There were variations and additional features in some of the assignments.

Splitscreen

The second principle which appeared often was an approach which divided the display of the device in two sections. For this approach there were also variations. The variations concern for example the representation of the images, there were assignments with cover flows and also some with images arranged in timelines or grids. The first screen had the purpose to present the collection of images, the other part of the screen served different purposes. In some assignments it served as a clipboard, in others it held the images in their new order.

4.2 Focus Group

A focus group was conducted to gain qualitative data for the design of an initial prototype. In the previous step items and categories were deduced from over 100 student assignments. Items were iteratively assigned to categories. The resulting categories and items were supposed to serve as a base for the focus group, which was kept in mind throughout the analysis. To keep the intro short and to stay within an acceptable time limit in total, a selection of the categories was made. Reducing the number of categories and items within the categories was supposed to prevent fatigue and the loss of focus of the participants while listening to the introduction. The task was explained in the introduction. This section is split up into three parts. The first part is the setup, which describes the structure of the focus group, including the tasks. The second part is the report, it reflects on what happened in the focus group. The last part, the outcome, explains the influence of the method on the succeeding work.

4.2.1 Setup

Five participants took part in the focus group. Everyone is accustomed to using a smartphone and also the camera of a smartphone. Every participant has a scientific background. The content of the focus group was based on the analysis from the submissions, which were discussed in Section 4.1. Audio and video was recorded to analyze the session afterwards.

Apart from the representations of images, which were deduced from the Interface and Interaction Design (IID) suggestions, three additional methods for representing Image Collections were added. Those three methods were elaborated in a seminar the author of this thesis participated in during the writing of this thesis. The seminar thesis was called "Time-Based Arrangement of Images inspired by Time-Dependent Data Visualizations" and can be found in Appendix A. The representation methods which were developed were called "Circle View", "Grid Circle" and "Anemone" and were incorporated in the intro representation of the focus group and the handout for the focus group. An explanation for the representations can be found either in Appendix A or Appendix B.

The focus group was designed as a workshop. At the beginning of the focus group a presentation was held, providing the participants with the information about the categorizations and items. Besides those information the procedure of the focus group was explained. The procedure was divided into the following six steps:

1. The first step was an introduction of the topic and the requirements, which stem from the use of a small screen and the need to be efficient and effective on small touch screens.

2. The second step was the presentation of the results from the review of the submissions (Section 4.1).

3. The third step was the task for each participant to develop an individual suggestion. The task for the participants was the same, like for the students from Section 4.1.

4. The fourth step was a group discussion of each suggestion. Each participant presented his or her concept, followed by a discussion with all participants.

5. The fifth step was a cooperative elaboration of one suggestion.

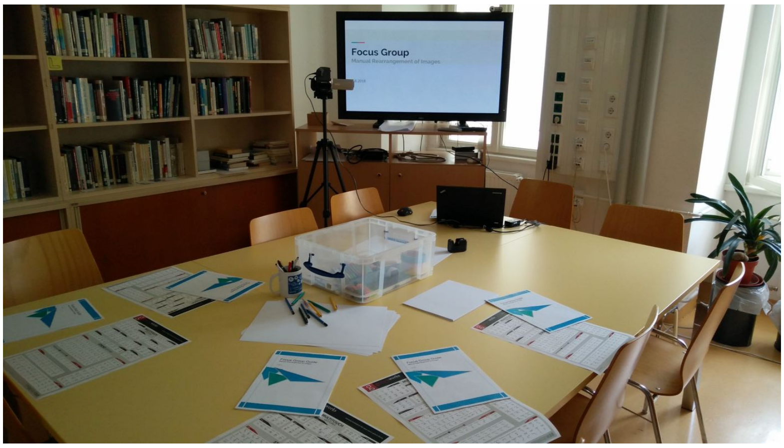


Figure 4.6: Setup of the focus group.

6. The sixth and last step was the discussion of the cooperative concept.

The instruction for the third and fifth step were the same as in the bachelor course "Interface and Interaction Design". Images shall be arranged manually on a mobile phone with a touch screen. The task is motivated by storytelling, only the storyteller knows the intended order of the images, therefore the task can not be solved by simply sorting the images according to their metadata, but metadata can be used to support the user. The example for the application was a collection of 100 - 300 holiday images which should be rearranged for the means of telling a story, which does not fit the chronological sequence. Not all images have to be used, the author can decide which images he or she wants to use. It was mentioned, that developed suggestions may include techniques, which have to be learned by the user at first, before the arrangement can be applied in an efficient way.

Each participant got a folder with the information which was presented at the beginning. The folder can be found in Appendix B. Material to create the suggestion was provided, this included for example different kinds of paper and pens and other material for tinkering. The work environment prepared for the focus group is depicted in Figure 4.6.

4.2.2 Report

The duration of the focus group was about 2 hours and 30 minutes. One concept per person was the result of the first half of the workshop. Each concept was explained by the participant. There was no time left to discuss special suggestions of the IID assignments.

After the first half of the workshop and before the instructions for the second half were given it was emphasized that the participants should lay more attention to the manual rearrangement of the images. This was pointed out, because some individual concepts neglected the manual arrangement and focused on group aggregations.

Different representations were chosen for the image collection. One concept built up on the anemone, the others used grids, although one participant experimented with different views, but also ended up with a grid view. The size of thumbnails were also discussed regarding the representation of the images. This was especially criticized in the anemone view.

Some concepts were split up into two phases. One example is the separation into *authoring mode* and *telling mode*. The authoring mode was further split up into a grid, a graph and tags of clusters. The grid contained the originals and supported the possibility to sort by metadata and tags. This was supposed to ease the manual creation of a graph in which images can be arranged as intended. Another concept contained a phase in which every picture is passed through and a decision is made if the image should be kept or rejected, either in the grid view or in fullscreen. The approach works like the dating app "Tinder"[4]. The participant emphasized that images that are sorted out in the process will not be deleted but end up in a separate folder, decisions can be undone and images can be retrieved from there. In this regard the difference between "delete" and "hide" was discussed. Some concepts were based on a preprocessing step of the mobile phone for clustering. It was also mentioned that it would be nice, if the phone could create a suggestion of an image order, that can be altered by the user, to save time.

Group aggregations were a topic which appeared repeatedly in the concepts. Those aggregations existed as manually created folders, tags or by sorting based on metadata. Throughout the focus group it was also discussed to have three priorities in addition. Images can be marked as green, yellow or red, the user can decide if the image is of interest (green), not of interest (red), or if it should be kept to decide later (yellow). In one concept the grouping was based on the purpose the image should serve.

Grouping images was also discussed in the context of similar images. The focus group tried to solve the problem to cope with multiple similar images of one subject. For example if the user took 20 pictures of a sunset, the system should be able to support the user to decide the user's favorite. In this context it was also mentioned, that the images have to be seen in fullscreen mode, to make a thoughtful decision.

Multiple selection was included in the concepts in different ways. One participant mentioned that multiple selection should be supported by different actions. For example both a simple tap and swipe should be possible, preferably in a way that would not require choosing in advance, so the selection mode can change fluently.

Use cases were discussed. The importance to know, how many images will be kept in the final collection and how many images of the original collection have to be rearranged, was pointed out. For example if the collection consists of 300 images and only 40 images will be used, with 4 or 5 images getting a new position, it is recommended to use a positive approach, where users choose the images they want to use. If only 30 images of the 300 will not be used a negative approach is recommended, where the user sorts

[4]https://play.google.com/store/apps/details?id=com.tinder, last accessed: 04/13/19

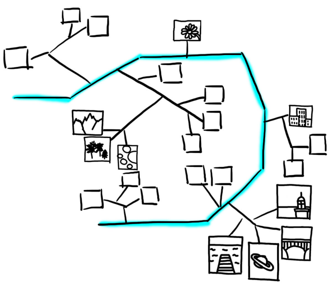


Figure 4.7: Conceptual image of the anemone.

out the images which should not be used. It was mentioned that it is important that the amount of images is reduced throughout the process. The more images were already processed, the faster it should come to an end, because fewer images are left to process.

One concept included the possibility to collapse groups and subgroups of the collection. The participant chose the anemone (Figure 4.7 and appendices) to represent the images, double taps on a branch collapsed the subtree. This accelerates the process by supporting the user to keep an overview of the collection and an overview of already processed groups. The idea of collapsing images continued in the second phase of the concept. In the first phase Images and groups of images can be added to the end of a timeline of images and removed from there. In the opinion of the participant, attaching images at the end would not be enough, therefore the images should also be rearranged in the timeline afterwards. Images can be collapsed between a start and endpoint, which are set by the user. It was stressed, that the concept builds upon fast gestures, therefore actions like touch and hold were not used.

Metadata was often mentioned in the presentation of the individual concepts, this included primarily chronological data but secondary also spatial data.

An approach similar to the splitscreen approaches from IID was suggested in the group discussion by one participant. The screen is split into two parts, which are both lists. One participant argued that drag and drop for adding the images in the other list should not be used. This was already mentioned before, because if the action has to be repeated multiple times it would take a long time. It was added that users have to remember which images they selected and where they want to move them, because the finger occludes a part of the screen while users are dragging the images. Enlarge images in the action of moving was suggested by one person as a possible solution. It was suggested to arrange images in the splitscreen approach by tapping on empty fields, where an image shall be inserted and immediately afterwards on the images, which shall be inserted at that position.

Approaches which build upon iterative principles were discussed. One participant suggested to highlight images that were already moved, which was complemented by another participant with the idea to highlight also images that were already viewed. One participant argued, that it would not be possible to create categories before all images in the collection were sighted. In the case all images are passed through, it is possible to reject some images that should not be used.

Throughout the group discussion video editing was brought up. A concept was developed, which uses multiple timelines, which are merged at the end. The timelines contain images from different categories. The images overlay images which are at lower timelines. It was further suggested to use the different timelines for different time periods, for example each timeline representing a different week. Another person argued that video editing software is based to a great part on taking over work from the user.

4.2.3 Influence on the Work

The data from the focus group was interpreted and some key points were extracted, which were aimed to be incorporated in the prototype.

No complex gestures were used in the suggestions, therefore no complex gestures are used in the prototype. The gestures that are used for the interaction should be fast, because they may have to be repeated multiple times for images. Therefore Drag & Drop is not intended to be used. It is beneficial if the paths for gestures are short, meaning if an image is dragged the distance between start and end should be as short as possible.

The system should support the user with decisions between multiple images of the same subject. It should be able to indicate when more images belong to the same subject, for example 14 images of a sunset. Furthermore it should incorporate the function to view the images in fullscreen, meaning the user can have a closer look at each candidate to choose a favorite among a set of images.

The user should be able to manually collapse images. This results from the possibly high amount of images in the collection and to support to keep some overview of the images. Collapsing also allows the user to decide when he or she is done with a group of images and to hide them somehow.

Furthermore, images which were already processed by the user should be indicated. This helps the user in keeping an overview over the progress.

It is of advantage if the system supports different selection modes. The implementation of multiple selection modes offers quick selection of images and meets different user preferences.

If images are “deleted” they should still be accessible in some way. For some users it is hard to erase images permanently. The possibility to retrieve removed images is a compromise between leaving them in the collection, although they are not needed, and removing them completely.

In the presentation of the anemone approach it was mentioned that it will not be sufficient if images are always added at the end of the current timeline.

4.3 Low-Fidelity Prototype

There was an intermediate step between the focus group and the interviews. The time was not only used to prepare the interviews, but also to create a LoFi prototype. The prototype was supposed to serve two purposes: the first purpose was to make some decisions about some characteristics and the second one was to talk about them in the interviews subsequently.

The focus group yielded insights about requirements and aspects that are considered to be useful. However, it also raised a number of alternative options for some functions or just requirements without the suggestion of solutions. Therefore this step was introduced to explore options and possibilities to incorporate wishes that were raised in the focus group.

In the analysis of the focus group session two possibilities of display of the images were considered to be suited to enable the incorporation of a number of requirements. These two alternatives are the anemone and the splitscreen. The anemone was discussed in the focus group. The point of criticism was the assumption of space inefficiency. The size of thumbnails was also discussed. The smartphone would not offer enough space to display an appropriate number of images at a time. The anemone on the other hand offers also advantages, like the collapsing of branches for keeping an overview while working with it. The decision between the anemone and the splitscreen was essential, therefore a screenshots of each view of the according alternatives were made, to make the decision. The term screenshot is used not for an actual snapshot of the screen, but for images, which were created with an image editing software to mock screenshots of the future application. Those were examined on a smartphone to get a realistic impression. The size of the screenshots was chosen to meet the display size of a OnePlus 3T [71]. The OnePlus 3T has a full HD resolution with an aspect ratio of 16:9, this corresponds to a resolution of 1920x1080 pixels. The display size measures 5.5 inches. The pixel density is 401ppi.

Although Hürst et al. investigate the thumbnail size, they neglect the pixel density of the device and state thumbnail sizes only in pixels and centimeters instead [42][41]. Therefore different sizes of thumbnails were tested for the mockups. The two concepts are depicted in Figure 4.8 and 4.9. The images were used to decide, which of the two concepts should be developed further.

The mockup of the anemone showed how efficiently the images have to be arranged to make good use of the available space. Some thumbnails were kept small on purpose, to investigate the size in which an image is still recognizable. The arrangement of the images in the anemone is also depending on the complexity of the trees and subtrees. The use of the anemone would require a complex algorithm to arrange the images. The algorithm needs to take into account the amount of images in a subtree as well as the surrounding subtrees, to evaluate how each branch and how each image is positioned ideally. Besides the variables the size of the images is important and does not only depend on adjacent images but also on the shape of the timeline. The shape of the timeline which was chosen

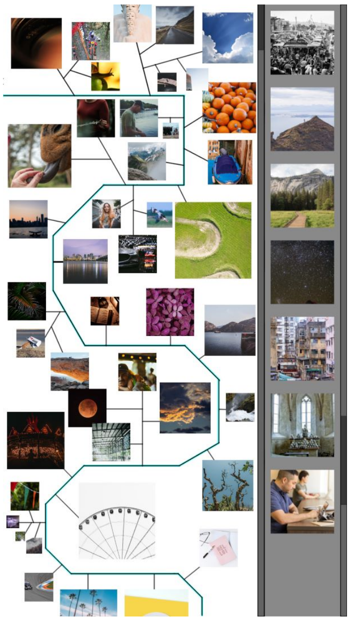


Figure 4.8: Digital draft of the anemone approach.

in the mockup limits the size of images at the left and the right borders. Some images are overlapping, which could cause confusion. The right part shows the chosen selection and order of images in a vertical timeline.

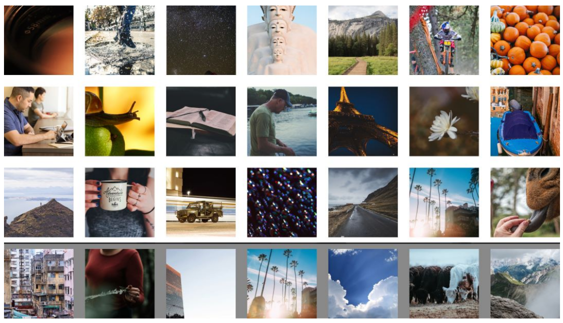


Figure 4.9: Digital draft of the splitscreen approach.

Figure 4.9 shows the screen of the splitscreen concept. The screen was horizontally split in a ratio of 3:1. The upper part represents the collection of images, the lower part represents the timeline of selected images in the order intended by the user. The parts are optically separated horizontally. The impression of the splitscreen is tidier than the one of the anemone view.

The splitscreen also makes it possible to arrange images quickly without loosing as much space as the anemone. Furthermore it appears less chaotic than the anemone. The need for an elaborated algorithm to arrange images meaningfully in the representation is a

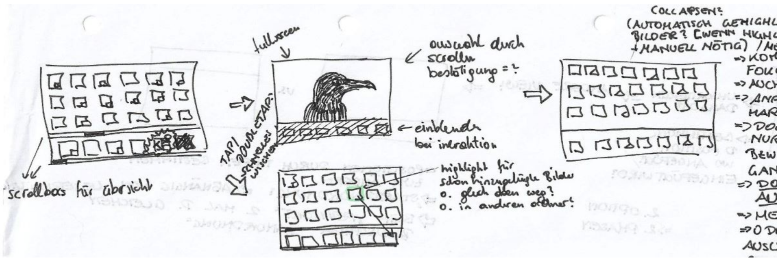


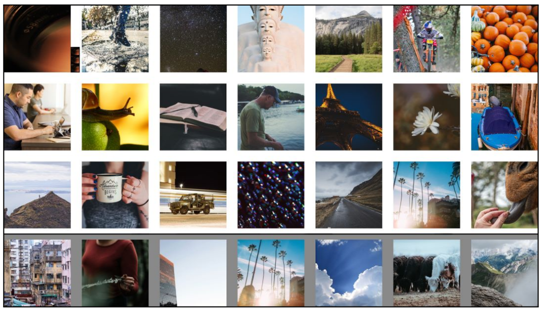
Figure 4.10: Sketches for prototype decisions.

big disadvantage. Therefore the splitscreen approach was chosen to continue with in the design process. It was further developed as prototype.

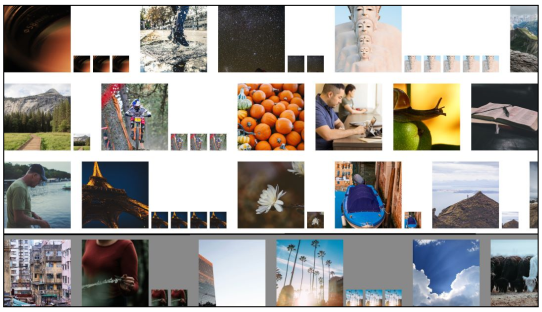
The prototype was supposed to support the decision making within the process as well as testing purposes and discussion basis with other persons. A digital mockup is better suited for some decision making than paper sketches. However, before creating the screenshots digitally with Gimp [5], the workflow was sketched and annotated on paper. Benefits and drawbacks were written on notes next to the sketches for decision making (Figure 4.10). After the workflow was thought through completely a list was written of the single steps of each function, that should be included in the mockup. For each step a screenshot was created. In some cases different approaches were created digitally to get a better impression of the alternatives and to help making decisions.

Figure 4.11 shows an example for the indication of alternatives of similar images. The first option (Figure 4.11a) tried to keep the regular grid, therefore the thumbnails of the alternative images are very small, also there is only space for one column of images. Because of the small size it was not possible to identify anything on the thumbnails. The thumbnails in the second option (Figure 4.11b) are bigger and were therefore preferred, but a lot of the space is wasted, therefore option three was created (Figure 4.11c). There is less white space in the third option, but, as in option two, the grid loses its regularity. If the images are traversed from left to right and top to bottom, this would be not much of a problem, but if the reading direction is from top to bottom and from left to right the images are not in one line. Therefore it was decided to choose a solution which only indicates the existence of alternative images for a subject, but not show the alternatives. Different sizes of squares and numbers were experimented with to evaluate the best combination.

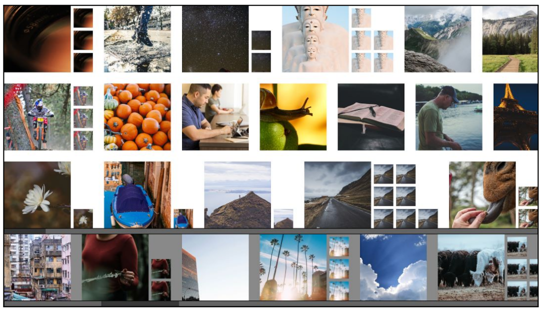
[5] https://www.gimp.org, last accessed: 11/18/18



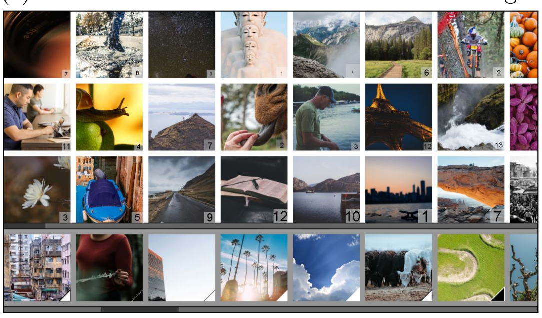
(a) Tiny thumbnails in a column next to the image.



(b) Thumbnails in a row next to the image.



(c) Thumbnails in a column next to the image.



(d) Numbers in a corner of the image indicate the number of alternatives of an image.

Figure 4.11: Possible approaches for the representation of alternative images.

4.4 Description and Workflow of the Prototype

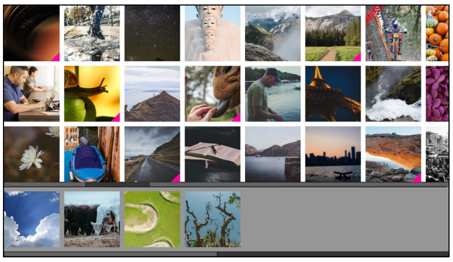
The function of the prototype was split into 7 functionalities: multiple selection, fullscreen, collapsing, move images to the upper screen, switch to the upper screen, position mode and the finesort phase. Multiple selection, collapsing and the position mode are to be activated and deactivated with buttons in the menu, which can be displayed with a swipe at the ride side. Each step of each function is represented by an image, the images are switched when a touch ends. Thus different gestures can be used to fake other gestures. For examples there is no difference between a swipe, a touch or a swipe with two fingers, because the action happens when the fingers lift up. As a precaution an area was implemented where a touch does not lead to the next screenshot, but to the previous picture. The screenshots are connected to each other. If one function was accomplished, the changes stay and the next function starts with that state. Changes stay till the end.

The starting point for the mockup is a collection of images on the upper part and a few images already in the timeline at the lower part. The screenshot is shown in Figure 4.12a. Pink colored corners at the right bottom indicate that alternative images are available. The *multiple selection* button in the menu is chosen, and a single image, as well as a group of images, is selected. The highlighted, selected images can be seen in Figure 4.12b. The images are flicked to the bottom and are therefore added in their original order at the end of the timeline. Images which are already in the timeline will be highlighted with a green frame. Images which were next to each other in the reading direction and were added to the timeline will be collapsed automatically. Because the reading direction is top to bottom and left to right the two collapsed groups will be created with 3 and 2 images accordingly. The collapsed images are marked by a pink colored corner at the top right corner of the group. Figure 4.12c shows the subsequent image after the images are added to the timeline.

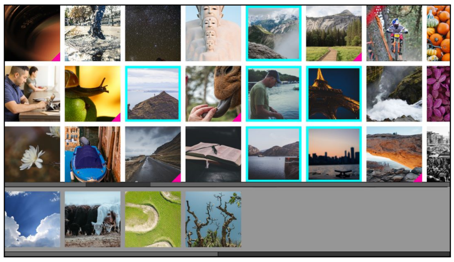
The next step in the mockup is the *fullscreen* view and the changing of an image to an alternative image. If one image, that is indicated with an pink colored corner at the right bottom is tapped on, then the image is shown in fullscreen. Swiping to the left or right opens a bar at the bottom with the alternative images. This is shown in Figure 4.13a. Another image can be selected by tapping on it at in the bar. The thumbnail will be replaced in the image collection by the new selected image, shown in Figure 4.13b.

When the collapse mode is turned on in the menu images can be *collapsed* manually. The according button has to be selected in the menu, to switch to the collapse mode. In collapse mode the first image is chosen and will be highlighted (Figure 4.14a), subsequently the last image, that should be collapsed will be chosen. All images in between will be stacked, behind the first selected image. The images are highlighted in green, because they were already interacted with and get labeled as collapsed image group, shown in Figure 4.14b. The images are highlighted as a reminder, because it is assumed, that with the action the user already made a decision for those images.

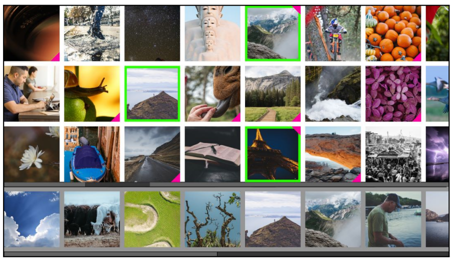
Thumbnails can be flicked to the top, to add them to a second workspace, which is intended to serve as a rejected or deleted image collection. The images are not deleted



(a) Screenshot of the application.

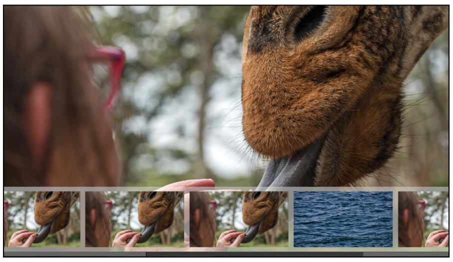


(b) Screenshot with selected images in a blue frame.

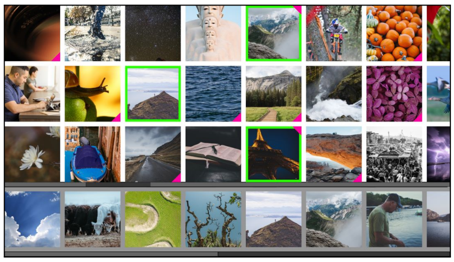


(c) Screenshot after adding selection at the bottom. Used images are highlighted by a green frame.

Figure 4.12: Process of selecting images and moving them to the bottom.



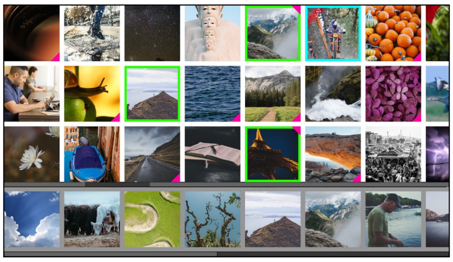
(a) Screenshot of fullscreen image with alternatives at the bottom.



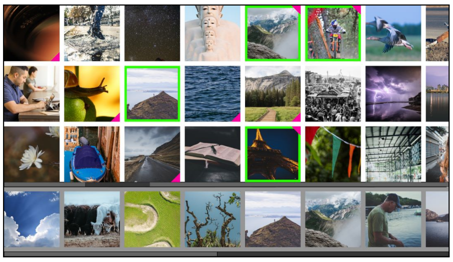
(b) Representative image of the group manually changed to the water.

Figure 4.13: Process of viewing images in fullscreen mode and choosing a representative.

because it was stated in the focus group, that it is hard to delete images and it would be favorable to have the option to switch to the deleted images. To cope with the high amount of images it should be possible for the user, to sort out some images, that will not be used. This happens by flicking one or more images to the top, then they disappear in the image collection, but they are not actually deleted. It can be switched to the collection of images that were sorted out by swiping with two fingers from top to bottom.



(a) First image of the collapsed group was chosen and is highlighted by a blue frame.



(b) Last image of the collapsed group was chosen and all images between the first and last image are collapsed and represented by the first image.

Figure 4.14: Process of manually collapsing images.

Figure 4.15 shows how the upper screen looks like, after four images were removed from the main workspace. The upper screen should have the same functionality as the main workspace.

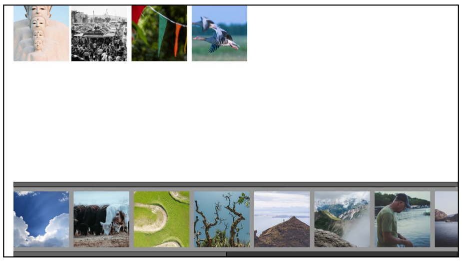
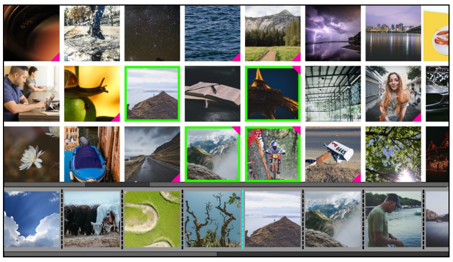


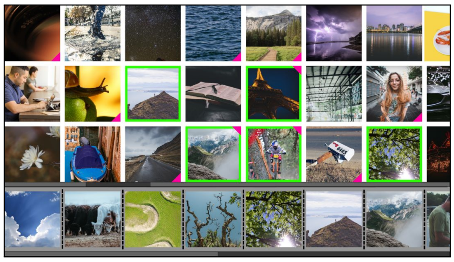
Figure 4.15: Upper screen of the application, serves as second workspace or deleted folder.

If users want to add an image at a specific *position* instead of at the end, where images are added per default, they can switch to the position mode via the menu. The position mode also enables the rearrangement of images, which are already in the timeline. The position mode displays dotted lines between images in the timeline. The dotted lines can be selected, by tapping on them, which highlights them (Figure 4.16a). If a position is highlighted one or more images (the position mode can be combined with multiple selection) can be selected which are added at that position (Figure 4.16b). Images can also be removed from the timeline, by flicking them from the timeline to the top.

When users are satisfied with the selection and preliminary order of the images, they can switch to the *finesort* mode. This happens by swiping with two fingers to the left. The finesort mode is significantly different than the previous mode. The screen is split in the middle and the timeline is shown twice (Figure 4.17a). The timelines at top and bottom are exactly the same, one change in one timeline does also impact the other timeline.



(a) Screenshot of the position mode with a selected position (blue, dotted line).



(b) Screenshot after an image is inserted at the specified position.

Figure 4.16: Position mode in the application.

The two timelines operate independently. This means, that in one timeline it can be scrolled to the end, while the other stays at the same position. Therefore images can be easily be moved from one end to the other. The reading direction in this mode is from left to right and from bottom to left, therefore images which are next to each other in the horizontal direction will be next to each other at the end. Positions can be displayed and used like before. Figure 4.17b shows a selected position after a dog image, selecting another image afterwards will move it to the according position. The change will take place at the top and bottom timeline.

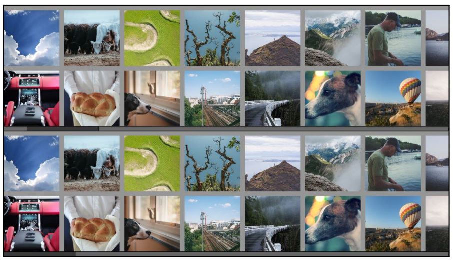
The mockup was presented on a smartphone (OnePlus 3T) during the interviews to present the functionality for discussion and feedback. It was demonstrated in the same order and mode, that was described.

4.5 Interviews

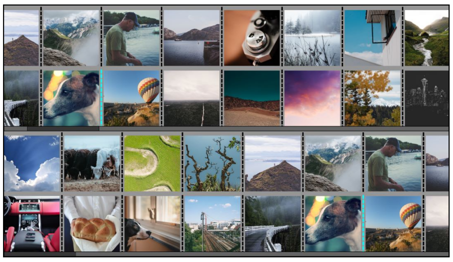
Interviews were conducted with three people who edit videos professionally. This section is split up into four parts: setup, data, discussion and impact on the work. Before the interviews were conducted one pilot test of only the mockup and one pilot test for the complete interview were done, to detect weaknesses or incomprehensibleness.

4.5.1 Setup

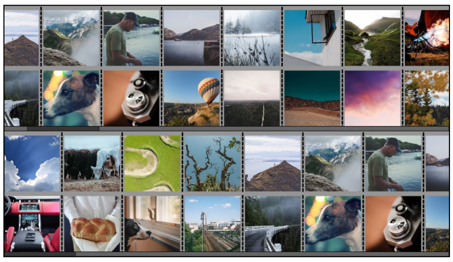
Three persons were interviewed about video editing. Everyone has a different viewpoint to the topic, which offers insights into three perspectives. Time and place were chosen by each interview partner. The duration was between 40 and 70 minutes. Two interviews were conducted in public in a coffee house, one was conducted at the university. Every participant was asked to sign a consent form, if they agree that audio is recorded, notes are taken and citations are made in the thesis. All participants agreed, therefore the audio was recorded and used in the analysis process afterwards. All interviews were held in German. The interview followed a prepared interview guide (Appendix C). The



(a) Finesort without position mode.



(b) Finesort with selected position in position mode.



(c) Finesort after image was inserted at the specified position.

Figure 4.17: Finesort in the application.

guide was split into two parts, the first part was about the participants and their work experience. The second part was based on the LoFi prototype, which is described in Section 4.3. The questions are listed in English below:

1. Thank them in advance for taking part in the interview.
2. Consent form.
3. About me, what I do, what the thesis is about.
4. Ask them to introduce themselves. Who they are and what they do in their job.
5. Why is sorting or ordering important in their job? What is the aim? What purpose does the ordering have? Are there any specifications from other persons or can they choose by themselves how they sort or order things?
    a) Which information is necessary? (Metadata)
    b) Which criterias do they follow?
6. What is their workflow? Which tools do they use for their work?
    a) What do they like about the tools?

   b) What do they dislike about the tools?

7. If they imagine to order images on a smartphone, how would their prefered workflow look like? Which purpose would they follow? Is the purpose different than the purpose in their work? Which steps are important to perform? (Overview, selection of relevant images, moving the images, looking for the new place they shall be inserted, fullscreen for the selection of images,...)
8. Does their work have an impact on their personal life if they think about the sorting or arranging of images?
9. Show, explain and discuss the prototype.
   a) Multiple selection.
   b) Fullscreen.
   c) Collapsing.
   d) Moveup.
   e) Upper screen.
   f) Positions.
   g) Finesort.
10. Potentially interesting topics for the mockup.
   a) Order / Aggregations / Pre-categorization step.
   b) Finesort / Postprocessing necessary? Does it make sense to have two or more phases in the process? Or do you think it would be better to do everything in the same way?
   c) Reading direction.
   d) Ratio between upper and lower part on the screen.
   e) The size of the thumbnails.
   f) Portrait or landscape format?
   g) Hiding or highlighting?
   h) Can you think of something (a functionality) that is missing to reach the aim?
      i. Insert all images from the collection in the timeline (or start with a different suggestion from the system instead of the empty timeline at the beginning)?
      ii. Redo?
11. Thank them again for participating.

The keypoints of the theoretical questions were talked about in every interview. The potentially interesting topics for the mockup were chosen when useful.

4.5.2 Data

The interviews were analyzed with the recordings of the interviews. Notes from the conversations were taken and keypoints were highlighted. Cohesive topics were connected within the interviews and also cross-linked between the interviews. Insights are enumerated subsequently, the enumeration is separated into video editing and the mockup. The items are marked by I1, I2 and I3, according to the interview partner who made this statement.

Video Editing

This section lists the outcomes from the interviews, which relate to the person, which was interviewed. The questions do not only approach video editing itself, but also questions, which are interesting to be answered from the perspective of someone, who has experience in video editing. In contrast to the mockup section, the questions were theoretical and did not include the introduction of an artifact. The participant is simply referred to as "P1", "P2", "P3" or "participant". All interview partners were male.

I1 Works as cutter.

I1 P1 usually does not know the files and the video footage in advance that he works with. Therefore P1 has to examine the files first, if the work task demands it.

I1 P1 mentions how important the folder structure is in the projects, especially in big projects.

I1 P1 mainly uses Adobe Premiere and Avid, depending on the employer. It is hard to tell advantages and disadvantages of these programs, because it depends on the projects. Avid for example is better suited for big projects, because it has a better server structure.

I1 There are two possibilities for the specifications for the video that will be created. Either a text is given from editorial staff or P1 has artistic freedom.

I1 Metadata like framerate, interlaced or progressive and the container format is important for P1 to be easy accessible.

I1 P1 asked for an example to reproduce the workflow, the answer related to exactly this example.

I1 P1 would prefer to avoid working on a phone and upload the files over dropbox to work with it on a desktop computer. P1 can not understand the purpose of a video editing app on a mobile phone.

I1 Concerns are expressed that it is hard to recognize quality flaws in the files on the phone.

I1 P1 likes to highlight or mark images as favorites that P1 likes.

I1 If P1 imagines an app on a smartphone, P1 would like to customize the interface.

I1 P1 has experiences with apps for video editing and criticizes the finding of specific functions.

I1 P1 mentions that tagging functions are pleasant in general.

I1 P1 prefers to file P1's images into folders at the beginning.

I1 P1 can not imagine a use case for an app to rearrange the images in a manual, not chronological, order.

I1 The file organization is something P1 thinks he also applies in private, because of the job.

I2 Is journalist and editor, constructs a dramaturgy. Thinks about the storyline before the video will be cut.

I2 Is familiar with the footage, because P2 is on-site when the footage is filmed. Although the perspective of the cameraman and the own one are not always the same.

I2 The video editing work happens together with the cutter, but P2 also has experience with the work as a cutter, because P2 used to work as one earlier.

I2 The storyline will be finished in parallel to the video editing work, the common thread is already known in advance.

I2 Often video clips are replaced and re-ordered trying different options for the final video.

I2 P2 states, that Final Cut Pro is different from the other video editing software, because it is somehow sticky and magnetic to move files around on the timeline. Everything is synchronized. This is comfortable to work with easy things. With more complex projects it is hard to break the "magnetism".

I2 P2 sometimes uses an video editing application on the iPad, also for professional purposes, but only on small projects.

I2 Avid and other video editing software are similar with preview window, a window for cutting and timelines. Therefore it is possible to orient oneself. Files can be dragged onto the timeline.

I2 The application on the iPad is simpler, it is limited to two tracks and has no effects, but it is good enough for many things.

I2 The interaction with a touchscreen is less accurate than the interaction with a mouse on a desktop device.

I2 P2 can not come up with an use case for manual arrangement of images on the smartphone. P2 usually just deletes unwanted images and the rest stays in a chronological order. Photobooks could be a possible use case, but it is also something P1 does not do.

I2 In the case P2 has to order images on a smartphone, the participant would imagine to touch and hold one image and drag it to the right position.

I2 P2 thinks the job impacts the dramaturgy in private. The participant supposes to think more about it, because of the profession.

I2 P2 takes a lot of images in private and thinks P2 could not find them again if the order would not be chronological.

I2 P2 would arrange the images thematically, for example all images of the Riesenrad, all images at the beach, but not only with people, but mixed with other images.

I3 Does video editing only on a amateurish level at work.

I3 Works only with short videos with a duration of a few seconds.

I3 Each video represents one word, the videos are ordered alphabetically according to the word in the video. The collection of videos are supposed to build a dictionary.

I3 Videos are stored in a database and additionally marked with tags, that add a theme to the word.

I3 The core functionality of Adobe Premiere works well, but there is a lot additional functionality, that makes the software more complex. Two screens with many windows and tabs inside them makes it unclear to him.

I3 P3 states that it is cumbersome to edit many videos at once with the same adjustment, because it is not possible with Adobe Premiere, because the participant can not find some function for it.

I3 Professionally P3 does not need any metadata except for the name of the file, because the participant names the files after the content. In private P3 needs the date, when the picture was taken, because the participant can deduce, the place from the time.

I3 It can be hard to mix images from two devices, because of different name conventions and sometimes also because of the timestamp the device adds.

I3 P3 would rename the files, to create a new arrangement of the images. If P3 would solve the task differently the participant would move the files. If there are 10 images at once the participant can start to rearrange those 10 images, for example move the image from position 10 to position 2. If there is more space from start to finish, a clipboard could do the trick to put images there temporally.

I3 P3 refers to the arrangement of application icons on a mobile phone. Moving images from one page to another, by dragging a thumbnail to the corner of the screen is unfavorable.

I3 Tagging is done in the database, not with a video editing software. P3 is not sure, if it is possible with a video editing software, but it would also be harder, because then it could only be edited by software that can read the tags.

Mockup

The following enumeration lists key statements from the interview, that relate to the mockup. As in the video editing paragraph the statements are assigned to the associated interview partner, who are again referred to as “P1”, “P2”, “P3” or “participant”. The mockup was presented to each interview partner. The initial situation of the first screenshot was explained. The interview partners were asked not to touch the screen, because the functionality is only mocked and not actually implemented, therefore it follows a pre-defined workflow. Some topics were prepared that are interesting for the prototype, in case the interview partner has nothing to comment.

I1 P1 asks, what happens, when the arrangement of the images is finished. The answer is it could be something like a diashow, to show the selected images in the new order to someone. This raises demands and questions about the diashow, like how long one image will be shown, what background music or visual effects between images will be available.

I1 Landscape format is good for the purpose.

I1 A nice-to-have would be to zoom in one of the timelines in the finesort independently of the second timeline.

I1 It would be too cumbersome to work with the images without a folder structure. From a specific number of photos on P1 would like to have a caption or a title, the timeframe, in which the images were taken and a representation image for a group of images.

I1 P1 would like to see highlights or favorites that were marked in other applications.

I1 P1 asked if the files are renamed to keep the new order, or if the files will be copied or if the metadata will be changed.

I2 The corners at the right top and bottom are confusing.

I2 The areas should be separated more clearly visually.

I2 P2 asks, why the thumbnails from the upper screen can not be dragged at a specific position in the bottom at the timeline, instead of adding them at the end of the timeline. Why they can only be set at specific positions in the position mode.

I2 P2 is skeptical if the different modes will not be confusing. It could happen, that someone forgets in which mode the participant currently works and makes a mistake because of it.

I2 P2 prefers to see the original format of each image, instead of only a square position. So the participant can also see at first sight, if the image is landscape or portrait format or if it is an panorama shot. P2 stresses how important the format is for him.

I2 If the menu is used very often P2 would make it available permanently. This is less prone to failure. It is annoying if you swipe to view the menu and it does not work.

I2 P2 thinks folders are not needed necessarily. The collapsing could also be used to create folders, but it would be good to add a title to find it again. It could also work on more levels and can be marked as something like “x2”. But the participant is not sure if folders are necessary at all.

I2 P2 thinks the landscape format is good.

I3 The upper and lower area of the screen should be separated more clearly.

I3 The corners and different color highlights are confusing.

I3 P3 likes the function to insert images at a specific position.

I3 The collapsing function is confusing.

I3 An intermediate step of deleting images would be nice. Where images are not deleted immediately but are in a clipboard, where they can be checked again, before they are deleted. This could undo mistakes.

I3 An animation of the actions would be good, to communicate the user what is currently happening, for example if thumbnails are moved.

I3 Folders could be of advantage. If there are folders P3 would need a title and additionally to a representative thumbnail of the folder.

I3 P3 would definitely also want to delete images, not only to hide them. Because some ugly images are not worth to keep anyway.

I3 P3 would prefer to start with an empty timeline, no suggestion of the system. P3 could not imagine what the system should suggest to start with.

I3 The decision about the reading direction is hard to make. Probably people would get used to it, but it should be marked. Scrolling up and down instead of sideways could be more intuitively, except for the decidedly timeline. P3 thinks it is strange to have images on the right or left that are hidden instead of below or above.

I3 A preview of thumbnails of alternative images of a subject could be shown, before it is switched to the fullscreen mode.

I3 There has to be a possibility to enter text, for example to save or find previously defined groups again.

I3 P3 is not sure, if the multiple selection is necessary. If the number of images in the collection is not that high.

4.5.3 Discussion and Influence on the Work

In this section the data of the interviews are interpreted and discussed. Statements are thematically summarized, independent of the person who made the statement. The composite statements refer to items identified by a letter. Overall observations are also taken into account and are discussed. An attempt was made to discuss keypoints which are of interest for the thesis. This was the case for topics, which already occurred in the focus group or before, or they occurred repeatedly in the interviews. Also topics which had a direct reference to mockup or existing subjects at this point. Topics which can be adapted from video editing to the prototype or which could be of interest at a later phase, were also considered as important and noteworthy. The items are classified as one of four loose categories: overall, organization, interface and functionality. The classification is only rough to provide an overview. Some items fit into more than one category. The categories are described shortly in the according section.

Overall

The section includes overall observations or statements. The items are of interest, but do not influence the remaining work directly at the current state. However, they may be important at a later phase.

A The initial point for video editing is a different one for each participant. One is not familiar with the material he is facing when cutting, the others are. In the first and second case there is a storyline, which directs the video editing process.
The users of the application will probably know the files, because they recorded them themselves. However, it is possible that images exist, that were retrieved from other people.

B In the first two interviews the order of the images follow a storyline, which is either given by another person or can be made up by the cutter.

C No interview partner could come up with an use case by themselves, in which they could make use of the manual arrangement of images.

D The preferred workflow on mobile phones for rearranging images manually did not include new interaction modes. The first interview partner was on a theoretical level closely related to video editing. It depended on the client from the given example. It made the impression, that the purpose of the thesis was misunderstood till the mockup part. It seemed to be unclear, that the aim of the thesis is to rearrange images manually, instead of videos, although this was mentioned in the introduction. The described workflows in the answers included drag and hold and a clipboard for images, which are farther apart. If they are close together they can be simply dragged around. They already occured in the suggestions in IID (Section 4.1). It should be mentioned, that the students and the focus group had more time to develop their suggestion than the interview participants.

E The mockup was operated in the same way as it is supposed to be the case in the prototype. Images are flicked to the bottom to add them in the timeline and to the top to move them to the second workspace. Switching between the workspaces happens by swiping with two fingers to the top or bottom, and a switch to the finesort by swiping with two fingers to the left. Unfortunately, this topic was not addressed in the interviews. Probably it was not recognized by the interview partners. This will be important for the prototype testing.

Organization

Organization includes items, which are related to how users keeps an overview over their files. It should ease the arrangement process by keeping the organization overhead as small as possible.

F In the first interview the importance of having organized files was mentioned several times, especially, when more persons work on the same project, but also in private. He prefers to put images in folders on the smartphone before arranging the images for private use. This was also something he stated in the mockup part, that it would be too cumbersome to work with the images without folders. Tags were also mentioned as a nice to have, as well as the highlighting or marking images as highlights. The second interview partner was unsure, if folders are necessary in the app. In the third interview file organization is necessary in the database and results directly from the term that is represented by the video. On a meta level videos get tags assigned, that describe a superior category. In the mockup part in the third interview the folder topic was also indecisive. Upon request it was stated, that it would be an option to convert the collapse function into folders. In this case it would be necessary to add a title, because it would be hard to identify a group of images with only one thumbnail.
Because the topic of folders is related to the collapsing function, it will be tried to combine both. This is discussed in item "G". Tags and marking favorites of images will not be included for the moment. Tags will complicate the arrangement process, because it would be necessary to add and delete tags, assign and remove tags to images, show tags and filter by tags. The effort for marking favorites would be a little bit less complicated, but for now the attempt is to keep the application simple. Tags and favorites could be topics, that are addressed in the testing of the prototype to figure out if this is necessary in the interaction to rearrange images.

G The collapsing function seemed to be overall confusing for the interview partners. It seemed to be unclear, that adjacent images are summarized into one, when they are added in the timeline below.
Like tags, folders would add complexity to the application while the goal is to keep it as simple as possible. However, the collapsing function appeared to be incomprehensible or not easy to grasp the way it was in the LoFi prototype, therefore some adjustments should be made for the prototype. The difference between the two colored corners was confusing at first for the interviewees. Although it is determined, that it is allowed to have elements, that have to be learned by the user, two colored corners for the difference between two different things seems to be unnecessary hard to distinguish. An approach should be tried to make the collapsing itself easier to grasp. Therefore thumbnails which are grouped together should get opaque to some degree. Furthermore, not only one image should be at top of the imaginary stack, but more images within the collapsed group should represent the group, for example four images which share the size of one regular thumbnail.

Furthermore, because it got mentioned more than once, that a title should also be used for folders, if folders exist, the collapsed group should automatically be annotated by the range of the date, beginning by the oldest image, ending with the date of the newest image. Although the size of one image divided by two in both directions will result in small thumbnails, it is supposed to communicate that there are more images at this position. In contrast to folders only images next to each other can be grouped together, but on the other side it prevents additional complexity, which would be the same case as with tags.

Interface

H It was confusing to separate the meaning of the corners at the right bottom and the right top. Also the highlighting, green and blue, is not clear at first sight.
This will be adapted to be more clear in the prototype. Replacing one of the corners and one highlight with something different could improve the clarity. The new marks should depend on the purpose, therefore the collapsing is discussed in more detail in the “Organization” section. For now, active thumbnails should keep a highlight in a noticeable color frame. The green highlights will be removed. This corresponded to already used images. Alternative approaches for the marking of alternative images were already tried, and the decision was already made once for the colored corners. Therefore the colored corner will stay as mark for the existence of alternative images. Although the color may be changed in the prototype.

I It was stated by two persons, that the upper part and lower part of the screen (the overview of the collection and the timeline) should be seperated more clearly. This also applies to the finesort mode, in which the timeline is shown two times.

J Two interviewees had the opinion, that the user could probably get used to the reading direction of the collection overview. It was discussed in more detail in one of the interviews. Advantages and disadvantages were mentioned of possible solutions, considering those, no solution can clearly outperform the others. However, it was stated, that it could be more intuitive to scroll up and down instead of left and right. In this solution the number of images in one row could be limited to a fixed number and the reading direction would be from left to right and from top to bottom instead of top to bottom and left to right. The latter one was chosen, because if one row contains an unknown number of images, the first and second row would not have anything in common. Adjacent images in the vertical direction could be apart weeks or months.

K The statement about showing and hiding the menu was also valuable. The thought which led to the decision to hide the menu was to save space for more images, but in case the menu is used often it could be annoying for the user, especially if it does not work at the first time, as the interview partner said. It is likely that one of the functions will be used often in the arranging process, therefore the menu should be shown all the time in the prototype.

L Concerns were expressed that the multiple selection, collapse and position mode could lead to problems, if the user mistakes the mode in which he currently is in. It could cause faults, therefore it is aimed to make clear somehow, which mode is active. In another interview it was expressed to animate some actions, to communicate to the user, what is going on. This falls into the same category, to make visible for the user, what is happening. The reasoning is comprehensible, the decision is left open for now, and will depend on further outcomes and the overall effort for other functionalities.
In the case, that the menu will be shown all the time, which is the aim for the prototype, it is easy to highlight the icon of the current mode.

M It was declared, that, before entering the fullscreen mode a small overview of thumbnails could be shown.
This could be favorable for some users but also disliked by other users. Therefore the colored corner will remain and the user will switch directly to the fullscreen mode for now. If this will be subject of discussion with other users it can be reconsidered.

N One statement when talking about the finesort was that zooming could be included. If one half of the screen could be zoomed independently from the second half it could be favorable. As with animations the reasoning is comprehensible and the decision is also left open for now and will depend on further outcomes and the effort for other functionalities.

Functionality

Items in this section deal with what the app can do and how it works.

O In one interview it was mentioned that it would be good to have an intermediate step, before images are definitely deleted. To have a view, in which the candidates for deleting are shown and to check them, before they are accidentally removed. Apart of the intermediate step the interviewee stated, that he definitely wants to delete images. There will be an upper screen, which can be used for sorting images out as well as second workspace. For now it is assumed, that unwanted images will be deleted in another gallery application. The app will only be able to create copies of images, but not to delete images.
This described functionality, which is quite similar to the upper screen in the prototype. Images can not only be flicked to the bottom, to add them to the end of the timeline, but also flicked to the top, to add them in another workspace,

which was originally thought of as a mix between deleting and hiding images. The user could move images to the top, he thinks he does not need or does not need currently. He could always move to the upper screen to move them back to the main workspace, because they are not actually deleted, but only hidden. The second workscreen should enable the functionality, the main workspace also has. Unfortunately there was not much feedback regarding the second workspace in the interviews. Regarding the actual deleting of the image there could be a recycle bin at the top of the second workspace. This would also fulfill the wish, to have an intermediate space, before images are definitely deleted. Furthermore users, who do not want to delete images can just ignore the function. On the other hand it may be tedious to switch to the second workspace to actually delete images.

P The question, what will happen after the images are arranged in the intended order was only raised once in the three interviews. Wishes were stated to adjust the time or effects and music in a diashow.
The focus of the thesis is the interaction for rearranging images and not how they will be presented afterwards. Therefore this part will be solved in a simple way, without any visual or sound effects.

Q In one interview it was stated, that the cutting work includes trying out different arrangements of clips and moving them around.
This should also be possible with the tested mockup. Inserting and removing images to and from the timeline should be easy and fast by using low-effort gestures. Also the position mode should offer a fast rearrangement within the timeline to try out some alternatives.

R Two interview partners mentioned apps for video editing for mobile devices, they had different opinions about them. One insisted on not using them and upload images on a desktop device to work there. It was too tedious to find the functions he was looking for and he did not want to invest the time to learn how to use it. He added that if anything, he would rather work on the iPad. The other interview partner said he would use an app on the iPad for small video editing projects. The application he uses is simple, it is limited to two timelines and a few effects, but he declares that it is sufficient for small projects. Also, as it was mentioned, detailed operations are not possible because the interaction with fingers is more coarse than the interaction with the mouse on a desktop device.
All criterions confirm to keep the prototype simple and do not add a lot of functions, which are not needed but make the application more complex and functions harder to find.

S In one interview the question was raised, why the thumbnails can not be dragged to a specific position in the timeline, but is added per default at the end. In case many images are added to the timeline individually it is faster, to just flick the images to the bottom, instead of looking for the right position between the timeline images. It is assumed, that the images are added to the timeline in the same order

as the image collection is browsed.
The possibility to always enable the user at which position the image should be inserted was reconsidered, but considering the criterias the decision is again made for adding the images per default at the end.

4.6 Iterative Development of the Functional Prototype

The functionality of the prototype varies only little from the mockup, which was introduced in Section 4.3. The functionality is still intended to encompass the following six features: fullscreen view, multiselection, collapse, upper screen, position mode and finesort. The development was split up into three milestones. The term milestone is used in project management. The development of a project is split into phases, the end of a phase is called milestone. In this thesis each milestone consisted of implementation and testing with users of the prototype. The results of the tests were analyzed and considered for improvements. Changes were incorporated in the next milestone. This approach was supposed to ensure the progress of the prototype moves into the right direction. The functionality, the test and the impact on the work is described in the following section for each of the three milestones. The final prototype is described in Section 4.7.

There were no requirements for test participants, beside being familiar with smartphones and touch gestures. It was tried to pick participants with different professional background and a approximately even percentage of female and male testers. The age of all participants were above 20 years, however, no such data was collected about the participants for statistical purposes. Participants in this section are referred to as “she”, independent of the actual gender. Notes were taken during user tests in each iteration, audio was only recorded in the last iteration. No difference is made between “swipe” and “flick” in this section, both refer to a linear swipe into one direction.

4.6.1 First Milestone

The functionality of the prototype after the first milestone encompassed the following features: multiselection, collapsing and fullscreen view including alternative images for a subject. The start screen of the first milestone is depicted in Figure 4.18.

Functionality Implemented

The mentioned functionality, which was existent at the prototype at that time is described in more detail subsequently. Besides the three functions, moving images from top to bottom and vice versa is possible. This is the core of the application and necessary to reach the goal. Moving images between top and bottom was performed by swiping images up or down. The image at the origin of the swipe gesture was moved up or down, according to the direction of the gesture. If the target was the storyline, the image was added by default at the right end. If the target was the collection, the image was added to the original position. Details to the other functionalities:

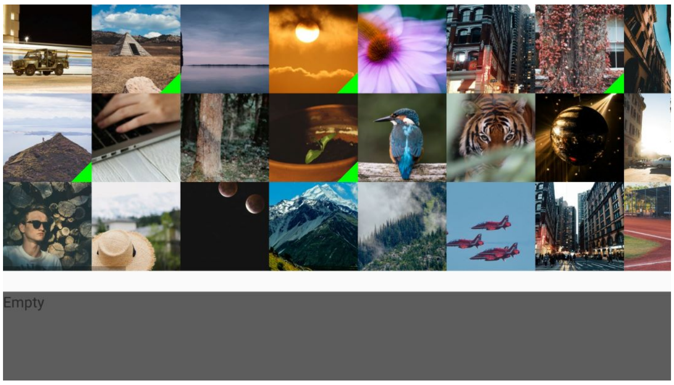


Figure 4.18: Prototype screenshot first iteration.

Move Images. As an essential part of the app, the implementation of how images are moved is discussed here. Images can be moved in the *Rough Sort View* between the collection (at the top) and the storyline (bottom view element of the rough sort view, see Figure 4.19 in Section 4.7). Flicking images in the collection to the bottom moves them to the storyline. Images, which are moved from the top to the storyline are removed from the collection. Images which are moved from the storyline back to the collection by a flick towards the top are removed there and inserted at their original position in the collection.

Multiselection. It was originally planned to provide a multiselection mode, which can be switched on and off with a button. If the multiselection mode is on, every image that is touched in a way, will be added to the selection. This does not limit the selection to just tapping single images, because images can also be selected by "drawing" a line to select multiple images at once. Within the development process it got clear, that this solution is not feasible, for technical reasons. Swiping left and right does also trigger a horizontal slide on the image collection. Therefore one gesture would be used for two different actions, which is technically not possible. Two options were taken into consideration to resolve the problem. The first was to lock the screen and disable scrolling, while being in the multiselection mode. The disadvantage is that it could get tedious to switch the multiselection mode on and off multiple times to scroll back and forth, while performing the multiselection. The other option was to lock the screen with one finger, while selecting the images with another finger at the same time. This corresponds to an analogy of holding something, to prevent it from moving. The decision was made for the second option. There will be no button for toggling the selection mode. The selection will be done by a two finger gesture, one finger which prevents the collection from scrolling and the other, which actually selects images, that are touched. This approach was chosen, because it is assumed to work more fluently than unlocking the mode each time, a user wants to scroll through the collection.

Collapsing. Due to the omission of the multiselection mode button it was considered to also replace the collapsing button and use something different to achieve collapsing. The principle of collapsing a number of selected images to a group was hardly commented in the interviews, nevertheless it made the impression to confuse the interview partners. In the mockup a screenshot with blue highlighted images was replaced with a screenshot with an image with a green highlight and a colored corner, additionally the other highlighted images disappeared, because they are stacked behind the front image and the other images moved closer to take their places. This is a big difference between two subsequent screenshots, which is hard to recognize. It was decided to not collapse images automatically if a group of adjacent images is added to the storyline, to prevent confusion. The highlights of selected images are a blue transparent veil above, instead of only blue frames, like it was in the non-functional mockup. That highlight should make the difference between different indicators more clear. In the first milestone collapsing was implemented by selecting multiple images and double tapping on one of the selected images. This also makes it possible to collapse a group of images, that are not adjacent. This also raised the question, which image will represent the group per default. It was chosen to take the image, which is double tapped for collapsing, as representative of the group. Furthermore, by creating a collapsed group by a double tap, it seemed obvious, to revert the action in the same way and expand collapsed groups by a double tap on the respective image. In this version the group of collapsed images as well as automatic alternative images are marked by green frames or pink frames, respectively. Therefore further opinions of the highlight can be gathered. Tapping on a collapsed image opens the group in the *fullscreen view*.

Fullscreen. A single tap on an image opens the fullscreen view. The fullscreen view first only shows that image. When the screen is tapped somewhere in fullscreen mode, a back button and a bar with alternative images at the bottom fade in. All alternatives of the fullscreen image are shown in the bar at the bottom. The alternatives were either detected automatically at the start, based on a close chronological difference of the date and time the images were taken (green corner), or they were manually collapsed by the user (pink corner). The time difference of images, which are automatically collapsed, were two minutes. A new representative of the group can be chosen by a tap on a picture in the bottom bar. The chosen representative is then shown in the collection instead of the old thumbnail.

User Test

The first milestone prototype was tested with three participants. The app was tested on the smartphone, which was used for development with a test dataset. The test dataset consisted of 150 images with three different aspect ratios and resolutions. The sample images were downloaded with "Bulksplash"[6]. Bulksplash allows the download of a number of images from Unsplash[7], a platform for free images. Additional parameters

[6] `https://github.com/MehediH/Bulksplash`, last accessed: 02/05/19
[7] `https://unsplash.com`, last accessed: 02/05/2019

can be defined in Bulksplash to specify the downloaded images (e.g., resolution). 13 out of the 150 images in the test dataset showed animals. The participants were asked to add images in an alphabetic order, according to the depicted animal on the image. The application had at that point only the above described functions. Those functions were introduced and demonstrated to each participant at the beginning of the test. The users were free to ask questions, during the test. It was stressed, that the correctness of the task is not measured. Furthermore it was mentioned, that the application they are testing is only a prototoype, and errors may occur. They were asked to constantly say their thoughts, expectations, intentions and similar things aloud ("Thinking Aloud"), to reproduce the users experience and to possibly detect flaws. After the task was fulfilled, a number of questions were asked. A participant may not be asked all, but only a subset of questions. The questions were:

1. How did you like the dimension of the images (size, aspect ratio, shape)? How would you prefer the representation of an image?
2. What do you think about the highlights? Was it confusing? Why? Do you have any suggestions for improvement?
3. What is your opinion about
   a) the efficiency? ("efficiency" may be defined by the user)
   b) the handling of the app?
   c) the gestures used?
   d) the understandability?

The first two questions were intended to evaluate decisions of details made. The third question concerns the overall interaction with the application, which was the focus of the user test.

Results and Impact on the Prototype

The sensitivity of the gesture recognition was overall a problem. All participants had troubles, because they often moved images to the top or bottom accidentally, because only a small movement of a touch already triggered the movement of the images. This was adapted for the next milestone.

The task was cumbersome to fulfill with the current functionality, because the images had to be added in the final order to the bottom. The position mode was not implemented at that time. It was annoying for the participants to find an image, which would have to be added earlier, because it belongs between already added images.

It was mentioned by two participants, that they would like to see the alternatives per default, when they open an image in fullscreen, instead of tapping the screen first, to display them. This was changed in the next milestone.

The opinions about the highlighting image groups (collapsed and alternatives) with frames were positive, because it was still possible, to see the image. Although the responses were positive, the indications of image groups in the second milestone were mixed. Automatically created alternative groups were marked with a green corner, as it was planned originally, the manually created groups kept a pink frame as indicator. With this mixture it was possible to have a direct comparison for the participants.

It was irritating, that everytime the participant switches from fullscreen back to the rough sort view, the scrollbar jumps back to the start, instead of sticking to the last scroll position in the collection, when an image is tapped. This was fixed in the succeeding milestones.

The task was obviously cumbersome to accomplish with the functionality present. It has to be thought ahead, which image has to be at which position, to move the images in exactly that order to the bottom. It was mentioned, that the absence of a rearranging option in the storyline is inconvenient. This was a functionality, which was planned to be implemented in the next iteration anyway.

The grouping operations received positive feedback. The manual grouping as well as the automatic grouping was liked. This verifies positive progress of the application.

The optical separation between the collection and the storyline was not salient enough according to the participants. That was less of a problem in the second milestone, but finally solved just in the third milestone.

The participants expected something to happen when they perform a long tap. It was reasoned after the test, that this gesture takes too long, if it has to be used for many images, and that gestures were used, which can be performed faster. It was suggested, to switch the multiselection mode on by holding an image for a few seconds, which is an established way for multiselection for touchscreen applications and people are used to it.

One participant was annoyed everytime she moved one or more images, because this caused succeeding images also to move to fill the position of the images, that are gone. Images used in the storyline were originally intended to be indicated in the collection, but this was not yet implemented in the first milestone.

Overall the participants were confidant, the app to be efficient, if it had less bugs. This referred to a flashing of the screen, which occured in two tests, and to the sensitivity of the swipe gestures to move images. The sensitivity of the swipe caused many accidental actions and it was not obvious for the user, what has happened. For such cases, the participants whish to have an undo button. The sensitivity of swiping to move images was reduced, to reduce such accidents. Tests of subsequent milestones will show, if the undo button will still be needed, or if that change is sufficient for now. The other gestures, that are used in the app were stated to be ok. It was posed, that it is a lot to learn at once and it was mentioned, that a help function would be of advantage. The help was part of the last milestone.

4.6.2 Second Milestone

The second milestone included the functionality of the first milestone (multiselection, collapse, fullscreen view and move images) and additionally position mode and trash view. New functions and adaptions of old functions are described next in this subsection, before the test and impact of the test on the prototype are discussed.

Functionality Implemented

This section describes deviations from the first milestone prototype. The functionality was either extended according to the original plan or adapted, based on the analysis from the user tests and considerations included.

Move Images. Images that are moved to the storyline do not disappear in the collection, but are indicated as used, by a transparent, light gray layer on top of the image. Furthermore images can not only be moved to the storyline, but also to the upper screen, also called the trash view. The trash view will be explained in more detail. Swiping images to the bottom, will add them in the storyline, swiping images to the top, will move them to the trash view. Furthermore the sensitivity of the gesture to move images was reduced, to prevent accidents.

Fullscreen. The alternatives and the back button are shown per default, when the fullscreen view is entered, instead of first tapping on the screen to display those elements on the screen.

Position Mode. When flicking images to the bottom, they are added per default at the end of the storyline. The position mode enables to select a position between images in the storyline, to add one or more images. Images can be moved in position mode either from the collection to the specified position or from the storyline itself. This enables the rearrangement of the images in the storyline. The position mode can be switched on and off by a double tap on an image, that is neither grouped nor selected. This works in all *action views* at the bottom and the top. Action views are the rough sort view, the trash view and the finesort view (described in the third milestone). Action views are highlighted in yellow in Figure 4.19 in Section 4.7.

Trash View. The trash view, former called the upper screen, is supposed to remove unneeded images in the collection. Images will not actually be deleted, because it is assumed, that this will be performed in other apps, but the possibility to move images to the trash may support the user to keep an overview. Images are moved to the trash view, by flicking them from the collection to the top. In contrast to adding images to the storyline, the images actually disappear in the collection, when they are moved to the trash view. The user is able to move them back to the collection, by switching to the trash view, and swipe one or more images to the bottom. The images will be added at their original position. Swiping to the trash view is performed, by scrolling with two fingers to the bottom in the collection of the rough sort view. Switching back to the rough sort view, works by performing a two finger scroll to the top on the collection in the trash view.

User Test

The prototype of the second milestone was tested with three participants, whereby one of the participants took already part in the first milestone test, because it was interesting to get a comparison of the old and the new prototype. The task was the same, the dataset was slightly changed. Some images were replaced with others, the amount of animal images was again 13. The approach was the same like in the first user test. The participants were asked to say their thoughts aloud and a few questions were asked after the task was performed. Also the questions were the same like in the first test, but with a focus on questions regarding efficiency, control, gestures and usability.

Results and Impact on the Prototype

The position mode was overall well perceived. The feature is useful and makes the task less cumbersome. The images do not have to be added in alphabetic order, because the order can be altered anytime. However, some participants had troubles with the function. Test users sometimes wanted to move images with a flick, when the target position is selected and highlighted, but it was intended to add images with a tap in this mode. Nevertheless the intention of the participants were comprehensible, because images are also moved with a flick, if the position mode is off. In the third prototype, which is also the final prototype, images can also be added to the selected position in the according mode with a flick.

Also participants tried to view images in fullscreen with a tap in the position mode, but in the case a target position is selected, the according image will be added at that position, instead of open the fullscreen view, which caused trouble sometimes. The test users wanted to remove images, which were added because of such accidents. The position mode was only designed to add images and not to remove them, therefore, they had to switch off the position mode first, however the test users did not remember at first, how. This was not changed in the last milestone prototype, but it could be of interest for future work.

The colored corner as indicator for alternatives was well received and does not have the risk of confusion with selected images, which was mentioned by two participants. The indicators for manual and automatic collapsed groups should be preferably be the same, therefore the alternatives are marked with pink and green colored corners in the final prototype. It was mentioned, that it could be extended with a number, which shows the number of alternatives. This idea was originally thought of, but it was dropped, because the text size must be big enough to be readable. The text would not fit into the size of the corners currently, therefore those approaches are hard to combine and it was dropped again.

The participants liked the automatic grouping function, they were interested, which criteria was used. They would prefer, to use more than one image of automatic groups, which was not possible at that time. It was also mentioned, that they would like to be able to extract a few images from the group. Breaking up an automatic group was not implemented in the last milestone prototype, but it makes sense to consider it in future work.

The participants were not familiar with all of the gestures or the use of them, because they are not established in commonly used apps. All participants in the test mentioned, that it is a lot to remember at first. Two times a help function or FAQ was suggested to add, to resolve this problem. A help button was added, which shows all possible actions on request.

As in the first iteration the participants were curious about long touch and drag and drop gestures, because they are used to some reaction on those actions. An explanation was given, why those actions are not used and the test users could understand the arguments. They suggested to use such gestures in addition. Also a pinching gesture to zoom and scale the thumbnails bigger or smaller was mentioned as a nice to have. The app and therefore the user could profit from those features, because smartphone users are used to move images with drag and drop or to have additional information or options with a long touch on an item, or scaling with a pinching gesture. However, none of the suggestions from this paragraph are implemented in the last prototype to put emphasis on the current interaction techniques further on, but they could be of interest for later, to make the app more comfortable.

Multiselection, collapsing and the trash were hardly used in this task, because the position mode eased the task. Sometimes the users also stated, that they forgot about the trash view. However, they can imagine, that those functions are valuable for other tasks.

4.6.3 Third Milestone

The third milestone was the final prototype. It contained all functions, which were planned and adaptions, which resulted from the user tests of previous milestones. The last milestone included the finesort view, besides the already described functions. Furthermore some minor changes or extensions were made, to make the app more appealing and some things easier to distinct optically. This included a black instead of a white background in the fullscreen view and a semi-transparent background of alternatives. This is supposed to communicate more clearly, that the images at the bottom do not belong to the other image and to show if there are more alternatives than can be currently be seen in the bottom bar. Images, which are used are colored darker in the collection, with the intention to seperate more clearly, which images are used. The divider between the top and the bottom in each view was colored in pink, to make it more salient. Following, the differences between the last and the previous prototype are described.

Functionality Implemented

In addition to minor optical changes, functional changes and expansions were made in the last prototype. Those resulted from previous user tests or were originally planned. The functions are listed below:

Finesort. The finesort view is an additional view, which offers the user the possibility to reposition images in his current storyline. It is intended to rearrange images easier, than in the rough sort view. Finesort presents the current storyline two times to the user. The storylines can be navigated independently, but a change in one storyline affects the other in exactly the same way. Therefore it is uncomplicated to reposition images over long distances, as soon as the user made a selection of the images she wants to use. It is always possible to switch back and forth between the rough sort view and the finesort view. Switching to finesort from rough sort with a two-finger scroll to the left, and vice versa with a two-finger scroll to the right. Finesort provides the fullscreen view and the position mode. It is a slimmed version of the rough sort view.

Help. To ease the acquisition of the interaction in the app a help function is provided. The help is opened with a tap on the help button in the right upper corner, labeled by a question mark icon. The help shows a graphic of the *action views* and labeled areas in the views at the upper half of the screen (Figure 4.19 in Section 4.7). Below the overview of the views all gestures are listed with a title, an depiction of the gesture, the description, the view and position, where the gesture has to be performed and the constraints to perform the action.

User Test

The user test of the final iteration was conducted on a larger scale, then the ones in between of the implementation phase. Seven participants took part in the user test. The test users had to be smartphone users, to make sure they are familiar with the overall concept of how to interact with such devices. There were no further requirements and no further data was collected. The test was designed to be closer to a real use case, therefore the participants were asked in advance to take a number of photos. The objects of the photos should meet a predefined short story. The story is about a nice day of a girl, who wants to remember the day with the help of images. The story was created in a way, in which keywords and important events can be depicted with everyday things. A few additional objects were listed, which do not appear in the story, to make sure to some spare images are present in the collection. The advantages of the task are: the participants did not have to come up with their own story, which saves their time and their approach can be compared, because they have similar images and the duration of the story is the same. The test users could focus on taking suitable images, and, because they took their own images, they are already familiar with the images in their collection. Therefore they already have an idea of what they are looking for when performing the task, which is supposed to be close to a real use case. The participants took the images with their own devices and brought them to the user test (i.e on a USB stick). Their images were uploaded on the test device at the appointment. This approach was chosen, because additional devices do not have to be handed out and the participants can take their own devices, which are often at hand anyway, and the fulfillment of the task can be observed. Questions could be asked or answered during and after the task in a conversation. Participants were again asked to think aloud. Pictures of two objects were, as stated in the guide of the task, already provided on the phone. It was stated, that images could also be downloaded or photographed from another screen, to keep the preparation effort for the participants low. It was considered to expand the task with special cases to cover all functionalities of the prototype, but it was dismissed, because of additional complexity and length of both the story and the task. The story could be looked up during the test to solve the task. The task and the procedure of the task was written down in a document and was handed out to each potential participant, before they agreed to take part. Since the native language of all participants was german, the guide for the test and the consent form was written in German, they can be found in Appendix E and Appendix F, respectively. The images were deleted as soon as the user test was finished. Audio was recorded to analyze the test sessions in case of taking notes was not sufficient.

The overall aim of the test was to help answer the following questions:

1. Is the provided scope of functionality sufficient to perform storytelling tasks?
2. Are the implemented interaction techniques learnable?
3. Can the implemented interaction techniques be fluently applied or are they too complicated? Why not? Why are they too complicated?

Answers for those questions should be identified with observations, thinking aloud and a short interview afterwards with the following questions:

1. What is your opinion about
    a) the efficiency? ("efficiency" may be defined by the user)
    b) the handling of the app?
    c) the gestures used?
    d) the understandability?
2. Is there something that could be improved in your opinion?
3. Is there something in particular that you really liked?

After conducting the first three tests, two small adaptions were made in the code. The timeframe for grouping images automatically was reduced from 2 minutes to 30 seconds. The timeframe of 2 minutes was chosen for real use cases, but images in this task were taken closely after each other. Subjects for this task were chosen to be easy accessible in each household, therefore it was possible to take pictures of everything within a short timeframe. The second adaption was to reduce crashes of the app. However, there were still situations, which caused the app to crash, but the participants were prepared for such cases and kept calm during the test.

Results and Impact on the Prototype

No further iterations of the implementations were made after this tests. Therefore the outcomes of these tests are recorded for discussion and future work. Outcomes include points of critique and things, which were noted positively by the participants. Possible improvements and adaptations, which result from the user feedback, are suggested. Answers to some of the interview questions were often anticipated, because matters were uttered already by the user thinking aloud. All observations and statements are listed below, without differentiation between the practical part and the theoretical part.

Participants again remarked, that it is a lot to remember at the beginning. At the end, no one doubts, that it is possible to learn and remember all gestures and functions to interact with the application quickly. How to switch the position mode on and off was often not remembered. But it could be observed, that after doing it one time, they were able to repeat it anytime, which is considered as a good indicator for the learning ability.

More than half of the participants drew on the help function. For one participant it was really easy to find, what she was looking for. She stated, that she remembers vaguely what she has to do and was therefore able to look for the according gesture icon in the list. Others had more troubles to find something particular. The diagram at the top was often neglected, which led to confusions regarding the list entries. One participant mentioned she likes the overview, but it is a pity, that it takes half of the screen space. The help function might need special attention to design it in a way, all people can comprehend.

Functions like multiselection, collapsing and the trash view were used only rarely in the task. The participants read the story from start to end during the test and moved the images to the storyline in parallel. As a consequence of automatic image grouping the amount of images was manageable. There was no reason in general to move images to the trash view or to collapse images manually to clear the collection. Also, because the images were taken in a random order by the participants, they did not follow the same order like the story. Therefore there was no need to move multiple images at once to the storyline. Due to the approach to add images one by one there was also rarely the need to use the finesort view to rearrange images. Switching back to the rough sort view, often related to as “main view” or “main screen” by participants, sometimes caused problems. Sometimes participants could not remember, how to get back, it was mentioned, that a button would be nice, which always enables to go back to the main screen. Another issue was the detection of the two finger swipe, because the gestures have to be performed either at the bottom or at the top, but not in the middle of the screen. This is definitely something that should be adapted to improve the interaction.

Multiple participants stated, that it would be of advantage to indicate, where she is currently, because the rough sort view looks pretty similar than to trash view. Making clearly visible for the user, which view she is in, is a valid point. The difference between those and the finesort view is easier to distinguish. For consistency reasons it would be good, to always show, which view is active. This could be indicated by a small icon with three views, which schematically shows the three views like in the help function, with a highlight of the current view. Furthermore it was posed sometimes, that a transition between both views would be helpful.

Like in previous tests, some participants again tapped long on an image, to see what effect the gesture has. It was again mentioned, that it could be used in addition, for example to switch the multiselection mode on and off. However, the participants were happy with the interaction of the application. It was mentioned multiple times, that the app reacts well to gestures and the quick, fluent interaction. The implementation of the prototype is considered as efficient, because of fast gestures.

The automatic grouping was again overall well perceived, but it came with a disadvantage in this case. In contrast to the task of the previous tests the participants were familiar with the photos in the collection, because they took them themselves. When reading the story, they had one or more images in mind, they want to use, but some images were “hidden” in groups. Therefore green indicated groups had to be inspected if they contain a certain image. Although the timeframe was reduced from 2 minutes to 30 seconds, some images were still summarized, even though they do not belong together. People often wanted to use more than one image of a group, which is not possible currently. Automatic grouping was liked overall, but wishes were expressed, to make it more flexible. For example enable the user to extract one or more images from a group or let the user decide by which factor or option images should be grouped. This is another point, which definitely should be incorporated in the future. Possibilities should be offered, to enable the use of multiple images of a group. This could be facilitated in different ways, automatic groups should be possible to expand manually for example, like it is already the case for manual groups. Or, as requested, offer the possibility to extract specific images from a group.

Like it was stated once, a loading icon should be displayed, when the finished storyline is saved, because this can take some seconds, without the user knowing, if his or her input is processed. A person stated that it would be nice to save the storyline and load it, to continue working on it. This would be of advantage in any case, to make it possible for users to continue working on a storyline another time or to work at more than one storyline at a time, without overwriting the last attempt.

4.7 Final Prototype

This section describes the final, functional prototype. The full functionality will be listed in Subsection 4.7.1. Those functions correspond to the results of the user centered design approach. Outcomes of previous conducted research methods are incorporated in the final prototype. The development of the prototype was described in Section 4.6. The referred section discusses three iterations in the development, leading to the final prototype, which is the third milestone of the digital prototype implementation. That section also explains all steps between the mockup, which was used in the interviews, and the final prototype, including feasibility problems and user-test results. This section describes the final functionality thoroughly in Subsection 4.7.1. Details of the implementation of the prototype will be addressed in Subsection 4.7.2.

4.7.1 Functionality

This subsection addresses the functionality of the final prototype, including how the functions are performed by the user. The functionality resulted from design choices based on the applied research methods.

The full functionality is split into eight parts, each is described later in detail. The application consists of five views, which the user can switch between (Figure 4.19). Views with a light yellow background are called *Action Views*. Action views are split into a top and a bottom part, which are divided by a pink line in the application. The *Rough Sort View*, also called *Main View*, the *Trash View* and the *Finesort View* are action views. The *Main View* is the start screen and provides the user's image collection at the top, and the storyline at the bottom. The storyline is empty at the beginning (Figure 4.20). Images can be flicked to the bottom to add them to the storyline. The *Trash View* and the *Finesort View* are placed above and right of the main view. Those two views can be accessed directly by scrolling with two fingers into the opposite direction, as it is also the case for one finger scrolling the user already is familiar with. Switching back to the main screen works accordingly. Because of this the two views are closely placed to the *Main View* in the diagram. The *Help View* can be accessed with a help button, which is placed in the top right corner in all three action views. The *Fullscreen View* can be opened with a tap on a thumbnail in each of the action views.

Move Images

Moving images and thereby ordering them is a core function of the application. Basically images can be moved between three areas. The first area is the collection in the main view, the place where all images are located at the start of the application. The second area where an image can be is the storyline. The storyline is the lower part of the *Main View* and keeps the images in the order, the user decides and therefore the order, in which the images are saved at the end. Images can be added to the storyline by flicking them from the collection towards the bottom. When images are added to the storyline, they get a dark gray veil in the collection. In Figure 4.21 seven images were added to the

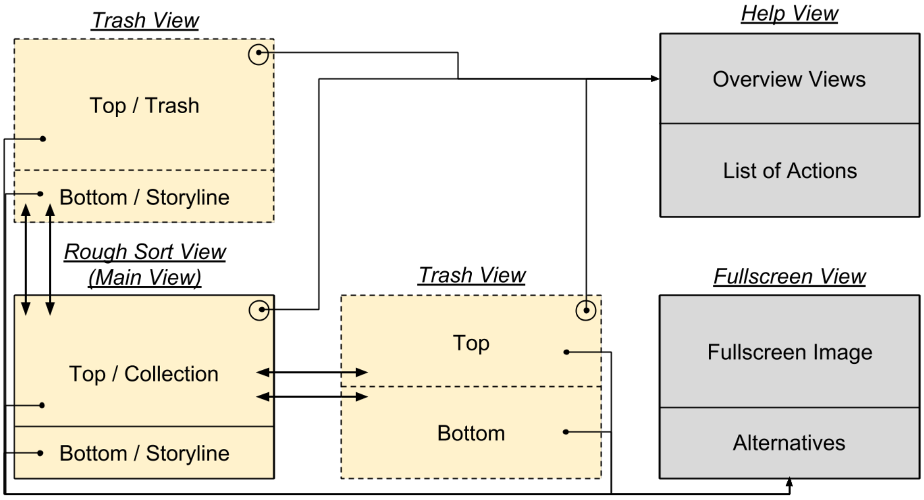


Figure 4.19: Existing views in the application. Action views are highlighted by a light yellow background. The italic, underlined text is the name of the view. The dashed or solid line presents the divider between the upper and the lower part. Non-italic, non-underlined text denotes the name of each part. Slashes are used for alternative names. Dots at the start of lines indicate a tap in that view element, arrows refer to the view that opens on that tap. Double arrowed lines refer to double scroll gestures to switch between views.

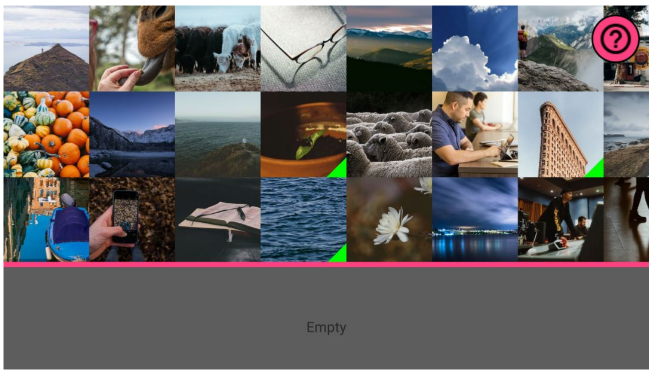


Figure 4.20: Start screen - main view (rough sort view) with the collection at the top, a pink divider and an empty storyline at the bottom.

storyline, the images are highlighted in dark gray in the collection at their initial position. Selected images can not be used again or moved to the trash. Images are removed from the storyline by flicking them towards the top. The image in the collection looses the gray veil, when it is removed from the storyline. Images are per default added at the end of the storyline, a different insert position can be chosen in the *position mode*, which is

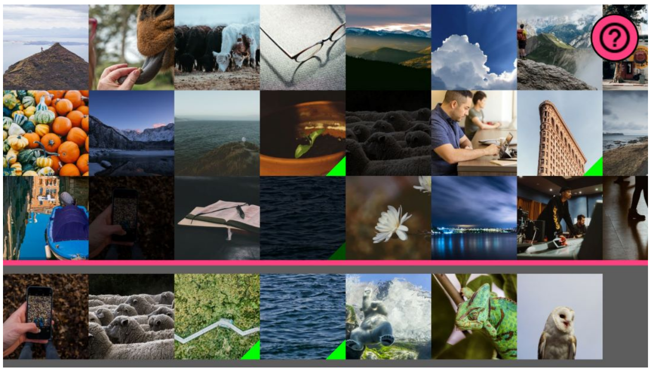


Figure 4.21: Seven images in the storyline, related images in the collection have a dark gray veil.

discussed later. Images can be moved from the collection to the trash by flicking them in the upper part of the main view towards the top. Images are removed from the collection instead of receiving an indication, when they are moved to the trash, but they are not deleted. The user can navigate to the *Trash View* and back by a two finger scroll.

Help

The application has a help function, to support the user in the handling of the application. A help button is placed in the top right in every action view. A screenshot of the help view can be seen in Figure 4.22. The help screen consists of two parts: the upper part always stays the same, it represents a conceptual depiction of the action views, and the label of each part. At the lower part of the screen is a scrollable list, which contains all possible actions in the action views. Each list item consists of the name of the action, an icon, which depicts the gesture to perform and the label of the part, where it has to be performed. A textual description is in the middle of each list item. Constraints for the action are written on the right side, for example "Position Mode Off.", furthermore the name of the view is written, where the action has to take place. The user can exit the help view with the back button at the top left.

Multiselection

The user can select multiple images at once and perform further actions with the selection. Actions might be to move more images at once or to collapse them, which is described later. There is no dedicated selection mode, which has to be turned on. The user can select multiple images by holding the according part of the screen with one finger and touching the images with another one. All images, which are touched are added to the selection and highlighted with a light blue veil. The images do not have to be tapped individually, although this is also possible. Deselecting images is performed analogously. If one finger holds the screen to prevent scrolling, a touch of a second finger deselects already selected images. Figure 4.23 shows a screenshot with five selected images.

Figure 4.22: Help view.

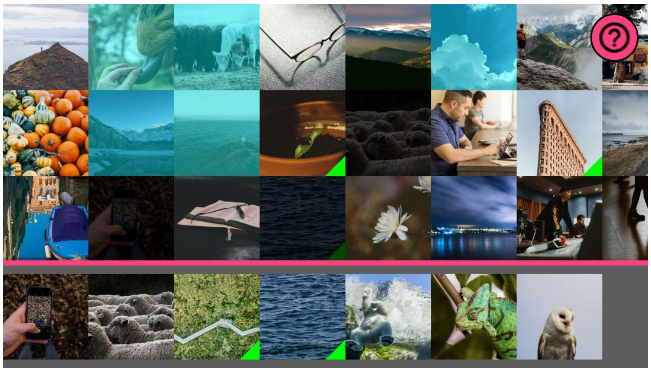


Figure 4.23: Five selected, highlighted images.

Collapsing

Collapsing is the aggregation of multiple images into one group. The group is represented by one image of the group, which can be chosen by the user. It is indicated with a colored corner to differentiate it from non-group images. There are two types of collapsing and both aim to support the user to cope with a vast amount of images in the collection. *Automatic* collapsing summarizes images based on the time difference of the date and time the picture is taken and is performed automatically when the app is started. *Manual* collapsing is explicitly performed by the user and can be applied in the rough sort view and in the trash view. Details to both types are stated below.

Automatic Images are grouped based on their last modification date or creation date. A threshold of two minutes is set, to summarize images, which have a time difference smaller than the threshold. Automatic collapsing aims to cluster images of the same subject. Per default, the first image is the representative of the group, but can be changed in the fullscreen view. Automatically grouped images are indicated with a green colored corner at the bottom right. The current representative is the only image, which is saved with the storyline, the alternatives within the group

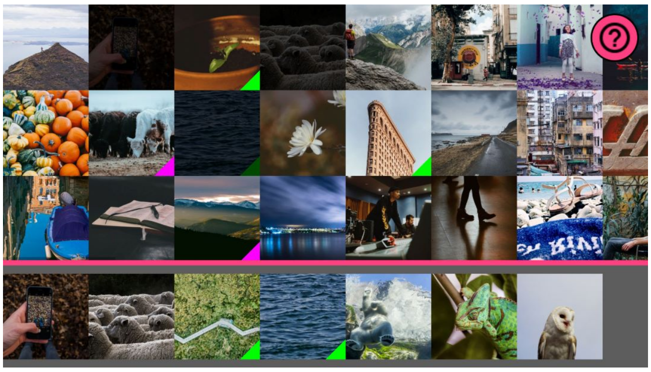


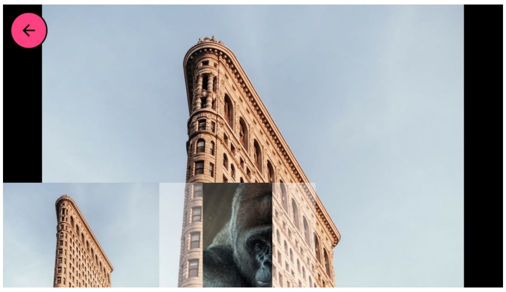
Figure 4.24: Screenshot with automatically and manually collapsed groups.

are not saved. An automatically collapsed group can not be expanded by the user. Figure 4.24 shows a screenshot with both kinds of collapsed groups.

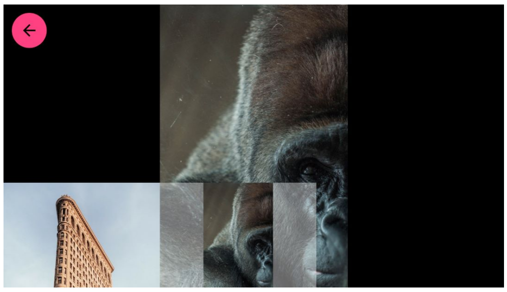
Manual Manual aggregations are created by the user and contain images, which were chosen by him or her. At least two images have to be selected, to create a group. If two or more images are selected (indicated by a light blue veil), one of the images in the selection can be double tapped, to summarize them into a group. The image, which was double tapped is set per default as the representative, which can be changed in the same way as in automatic groups. Manually collapsed groups are indicated by a pink corner at the bottom right. Two manual groups can be found in the screenshot in Figure 4.24. Manual groups can be expanded again, by a double tap on a collapsed group. Manually created groups can only be moved between the trash and the collection, but can not be added to the storyline.

Fullscreen

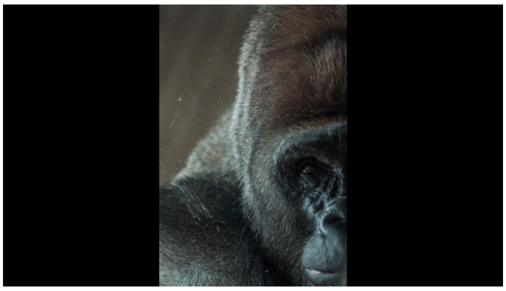
Because images are only shown as thumbnailsin most of the views, users can also open images in fullscreen. A single tap on a thumbnail in one of the action views opens the respective image in a fullscreen view. The fullscreen view shows per default alternative images of the current image in a bar at the bottom of the screen and a back button in the upper left corner, depicted in Figure 4.25a. Alternative images are images, which belong to the same group, whether they were collapsed automatically or manually. A new representative of the current group can be chosen by simply tapping on one of the alternatives. This changes the image in the background to that image (Figure 4.25b). If the background is tapped, the back button and the alternative images are hidden (Figure 4.25c), until the background is tapped again. The last picked image in the alternatives represents the group in the collection (Figure 4.25d).



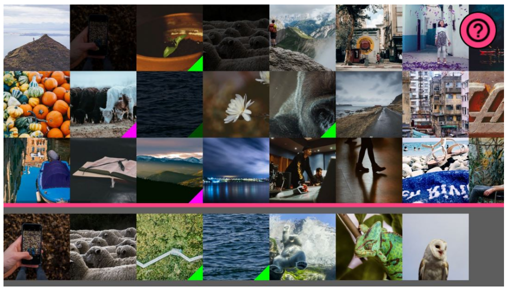
(a) Fullscreen view start.



(b) Fullscreen view after changing the representative.



(c) Fullscreen view background.



(d) Collection after representative change.

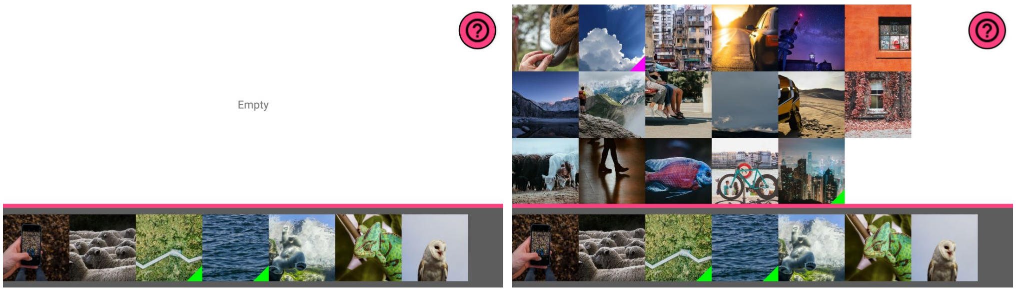
Figure 4.25: Fullscreen actions.

Trash View

Images can be moved to the trash with a flick towards the top in the upper part of the main view. Images are then removed from the collection and moved to the trash view. The files are not deleted, but sorted out from the collection. Thus, images can be recovered in the collection by switching to the trash view and flicking the images towards the bottom. Switching to the trash view is performed by a two finger scroll to the bottom in the rough sort view. If no images were added to the trash, the upper part is empty and only the storyline is shown (Figure 4.26a). Images are arranged in the insertion order. Single images and collapsed groups can be moved to the trash and vice versa. Images, which are already used in the storyline can not be moved to the trash. Figure 4.26b shows the trash view with a few thumbnails.

Position Mode

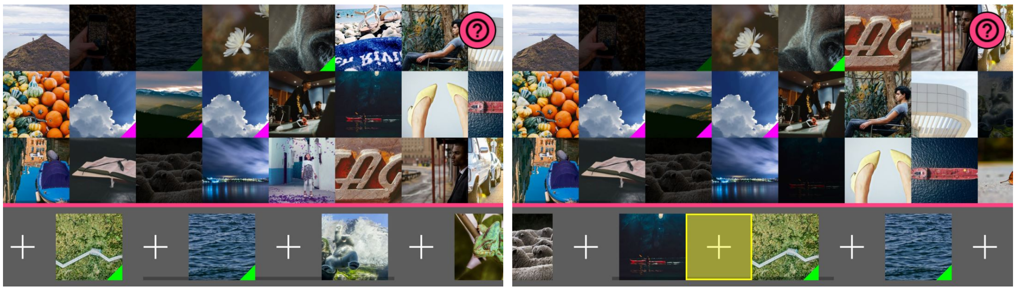
Per default, images are added at the end of the storyline when they are inserted. In contrast, the position mode allows to insert images at any position in the storyline. The position mode can be toggled on and off by the user with a double tap on an image in one of the action views. The tapped image may neither be a manual collapsed group nor selected. Figure 4.27a shows the main view, when the position mode is on. Placeholders with a "+" sign are at the beginning, the end and between all images, to indicate possible insert positions. The user selects an insert position by a single tap on one placeholder,



(a) Empty trash view. (b) Non empty trash view.

Figure 4.26: Trash view.

the placeholder will then be highlighted with a yellow veil, which can be seen in Figure 4.27b. Images, that are tapped or flicked to the bottom subsequently, after a placeholder has been selected, will be added to exactly that position. More than one image can be inserted at once or one after each other. The placeholder remains selected, until the user taps it again or taps another placeholder. In the position mode images can not only be added to a specific position from the collection, but can also be moved from one position in the storyline to another position. Therefore images do not have to be moved back to the collection, to shift them to another position.

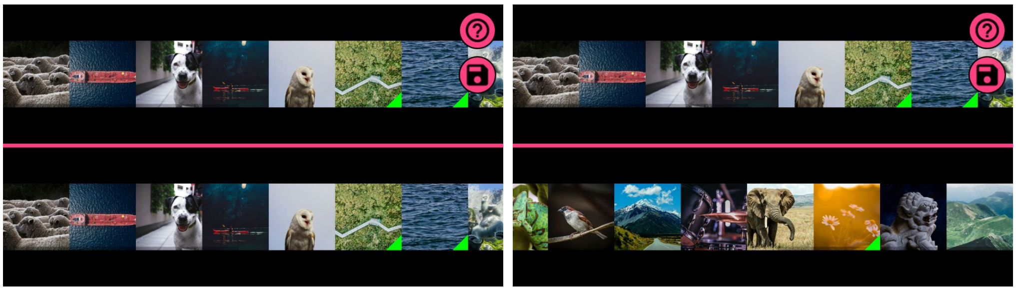


(a) Position mode on in main view. (b) Selected placeholder in position mode.

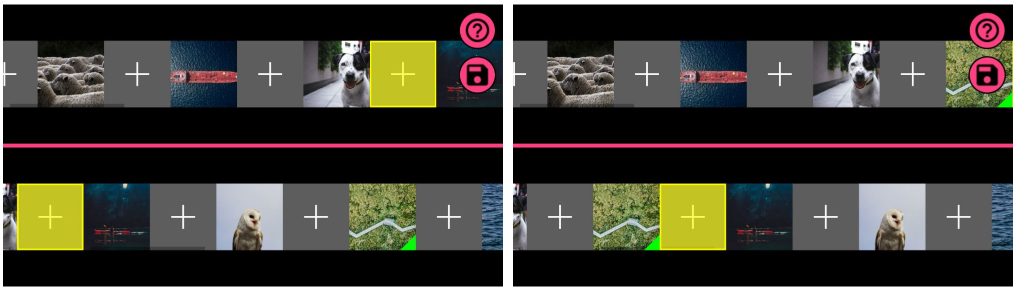
Figure 4.27: Position mode.

Finesort

The finesort view is for fine tuning the order of a selection of images in the position mode. The view is split in the middle of the screen, both halves show the current storyline (Figure 4.28a). The bottom and top storyline can be scrolled individually, therefore one half can be scrolled to the end, while the other is at the start or somewhere in the middle (Figure 4.28b). This is supposed to ease shifting actions over a long distance in the storyline. As earlier described, the position mode can be switched on and off with a double tap on an image. If one position is selected, it is highlighted in both storylines, depicted in Figure 4.28c. Changes always affect both storylines, because they are the same storyline. Figure 4.28d shows the same storyline, after the green image has been moved to the selected position. When finished, the creation can be saved with a press on the button below the help button. The user is asked to confirm the operation, before the images are saved.



(a) Finesort view. (b) Individually scrolled storylines.



(c) Position mode in finesort with selected target position. (d) Finesort in position mode after image shift.

Figure 4.28: Finesort view.

4.7.2 Implementation

This chapter discusses the technical specifications and the underlying software architecture in the respective subsections. This includes individual components of the application and how they are related to each other. The section concludes by illustrating an example of how user input is processed in specific use cases.

Technical Specifications

The application was developed with Android Studio[8] for Android devices. Kotlin[9] was chosen as programming language. A OnePlus 3T was used for the development, the code was mainly tested during the development phase with this device. Furthermore, this device was used in the usability tests of the three iterations. Code parts, which were

[8] https://developer.android.com/studio/, last accessed: 02/18/2019
[9] https://kotlinlang.org, last accessed: 02/18/2019

taken from external code samples, Android, StackOverflow[10], tutorials or similar, are annotated in the code to the author's best knowledge.

Software Architecture

This section covers the components and their functions in the application. It describes the structure of the code, including classes provided by Android and custom classes, implemented by the author. In the following section classes provided by the Android framework are referenced by their original notation and respective class names, for example RecyclerView.LayoutManager[11] . Classes, which were implemented in the course of this work are highlighted by using italic fonts, for example *FinesortActivity*. The software architecture is described aligned to the views, which exist within the application. Figure 4.29 depicts the responsibilities of the according activities for the views. All fragments existing within the views are depicted in Figure 4.30. As it can be seen, not every view consists of fragments. View elements which are either embedded in a fragment or directly in the view, which belongs to an activity is shown in Figure 4.31. When comparing Figure 4.30 and Figure 4.31 it can be seen, which view elements are attached to a fragment and which view elements are not.

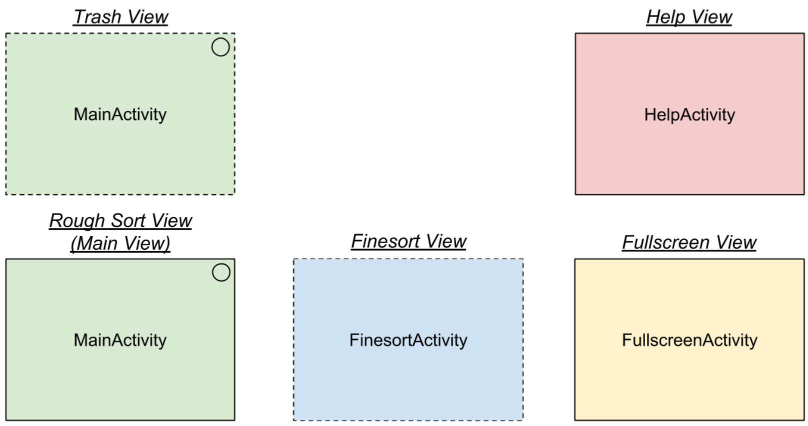


Figure 4.29: Activities of the prototype aligned to each view.

- ***MainView.*** The *MainView* is managed by the *MainActivity*, which is the activity, that is started, when the app is launched. The *MainView* consists of two fragments: the *GalleryLowerTopFragment* and the *GalleryBotFragment*. The latter one is responsible for displaying and reacting to requests regarding the storyline. The *GalleryLowerTopFragment* is one of two subtypes of the abstract class

[10] `https://stackoverflow.com`, last accessed: 02/19/2019
[11] `https://developer.android.com/reference/android/support/v7/widget/RecyclerView.LayoutManager`, last accessed: 02/19/2019

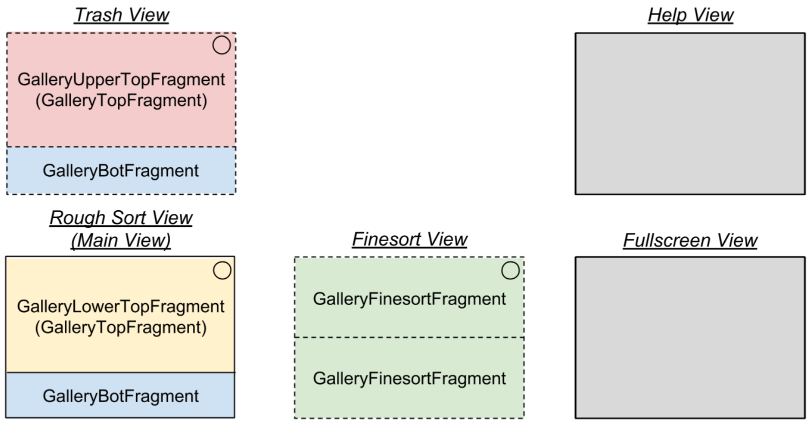


Figure 4.30: Fragments of the prototype aligned to each view. HelpView and Fullscreen-View do not have fragments.

GalleryTopFragment. The *GalleryLowerTopFragment* is responsible for viewing the collection of images, which are available for a user. The *GalleryBotFragment* displays the current storyline. User interactions in the *MainView* are either processed in the *GalleryLowerTopFragment* or the *GalleryBotFragment*, according to the area, where a gesture is detected. Both fragments in the *MainView* contain a RecyclerView[12], which handles the display and processes the user interaction. Instances of the RecyclerView.OnItemTouchListeners are attached to the RecyclerView to facilitate those functionality. A RecyclerView.LayoutManager arranges each image from the dataset either in a grid or in just one row. In this application only the subtype RecyclerView.GridLayoutManagers[13], which are a subtype of RecyclerView.LayoutManagers, are used. A RecyclerView.ItemDecoration[14] highlights specific images. RecyclerView.OnItemTouchListener[15] detect touch input and process them accordingly.

- ***TrashView.*** This view is also managed by the *MainActivity*. The view consists, like the *MainView*, of two fragments: the *GalleryBotFragment* and the *GalleryUpperTopFragment*. When the user switches from the *MainView* to the *TrashView* the lower fragment stays unaltered, but the *GalleryLowerTopFragment*, is replaced by the *GalleryUpperTopFragment*, which is the second subtype of the *GalleryTopFrag-*

[12] `https://developer.android.com/reference/android/support/v7/widget/RecyclerView`, last accessed: 02/19/2019

[13] `https://developer.android.com/reference/android/support/v7/widget/GridLayoutManager.html`, last accessed: 02/19/2019

[14] `https://developer.android.com/reference/android/support/v7/widget/RecyclerView.ItemDecoration`, last accessed: 02/19/2019

[15] `https://developer.android.com/reference/android/support/v7/widget/RecyclerView.OnItemTouchListener`, last accessed: 02/19/2019

ment. The latter fragment displays all images, which were moved to the trash. Like the *MainView* the *TrashView* uses a RecyclerView to display images and process user input. Again, RecyclerView.LayoutManager, RecyclerView.ItemDecoration and RecyclerView.OnItemTouchListener are attached to the RecyclerView.

- ***FinesortView.*** The FinesortView is managed by the *FinesortActivity*. The view consists of two *GalleryFinesortFragments*. They are sized equally and are built in the same way. One *GalleryFinesortFragment* consists of one RecyclerView to show one row of images. The images which are shown are the same images the *GalleryBotFragment* shows. The RecyclerView.LayoutManager for both RecyclerViews are a RecyclerView.GridLayoutManager and both have the same type of RecyclerView.ItemDecoration attached. The RecyclerView.OnItemTouchListener which belongs to the RecyclerViews has less functionality than the one in the *GalleryBotFragment*. The listener does not support multiselection and collapsing.
- ***FullscreenView.*** The *Fullscreen View* is handled by the *FullscreenActivity* and consists of one image, which covers the whole screen and, optionally, a RecyclerView with a RecyclerView.GridLayoutManager at the bottom, that displays one row of images. In contrast to all other RecyclerViews, the FullscreenView RecyclerView is not embedded in a fragment and it also does not have a RecyclerView.ItemDecoration. The RecyclerView.OnItemTouchListener that belongs to the RecyclerView only supports tapping gestures to select another representative of the current group.
- ***HelpView.*** The *HelpActivity* manages the HelpView. The view consists of one image in the upper half of the screen and a scrollable list of possible actions within the app in the lower half. The image is pre-defined and shows the action views of the app. The list contains all gestures, which can be performed in the app including the place, defining which view and which view element, and the effect of the gesture.

Figure 4.32 depicts the software architecture. The types of classes are color coded. The color codes support the visual recognition of recurring patterns. Fragments (red), always have a listener (green), a decoration (violet) and an adapter (blue). Those three components are needed for the RecyclerView in this work. The components which are depicted in the graphic are described below. The application consists of four activities. Activities may consist of one or more fragments. In one special case an activity (*FullscreenActivity*) does not have a fragment. However, it also consists of a RecyclerView with a listener and an adapter.

- **Activity.** An activity is a fundamental component in Android [25][24]. They provide callback methods for different kinds of requests. For example they handle user input or execute methods which are called from other activities or classes. An activity provides the user interface for the interaction with the user. The prototype consists of four activities, the *MainActivity* is the entry point for the user, the others can be accessed from there.

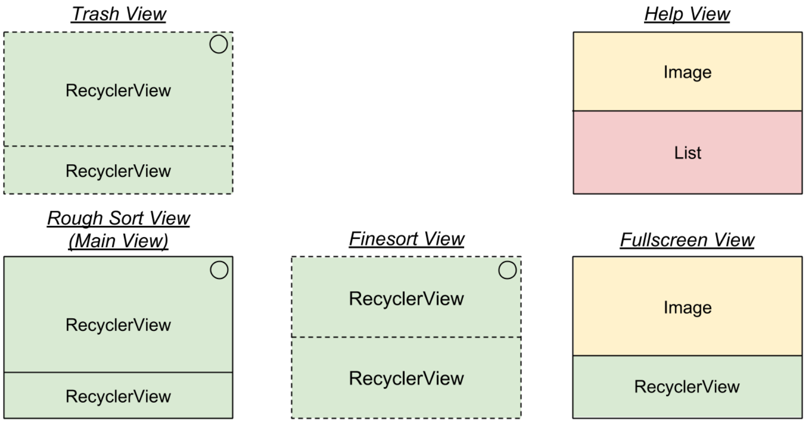


Figure 4.31: View elements of the prototype aligned to each view.

- ***MainActivity.*** This activity is started, when the app is launched. It consists of two fragments, a *GalleryBotFragment*, handling the storyline, and an abstract *GalleryTopFragment*. The *GalleryLowerTopFragment* and *GalleryUpperTopFragment* are of the type *GalleryTopFragment*. They are replaced by the activity on respective user input. The *GalleryLowerTopFragment* represents the Rough Sort View, the *GalleryUpperTopFragment* is the collection of the Trash View. The main activity is able to start any other activity. The majority of actions is processed in the respective fragment.
- ***HelpActivity.*** This activity shows possible inputs and actions, that can be performed in the application. The user interface of this activity consists of a graphic and a scrollable list of possible user input. The user can only navigate back to the previous activity from there.
- ***FullscreenActivity.*** The *FullscreenActivity* can be started from every Action View. In contrast to the other image views, this activity does not have a fragment. However, it consists of a RecyclerView, which represents images in the current group. A user is only able to navigate back to the previous activity.
- ***FinesortActivity.*** This activity is similarily structured compared to the *MainActivity*. It consists of two *GalleryFinesortFragment*s, one at the upper half and one at the lower half of the screen. The fragments are not replaced throughout the runtime. User input is mainly processed in those fragments. *GalleryFinesortFragments* are similar to the *GalleryBotFragment*, which is the storyline. Each other activity can be started originating from the *FinesortActivity*.

- **Fragment.** A fragment is a component, which is embedded in an activity [26]. It can be replaced by another fragment and enables to build a modular user interface.

The core of each fragment in this application is the RecyclerView. The RecyclerView needs an RecyclerView.Adapter and an RecyclerView.LayoutManager to work correctly. Each RecyclerView in this app has it's own adapter implementation, they are color coded in blue in the diagram in Figure 4.32. A GridLayoutManager is used as it is provided by Android as RecyclerView.LayoutManager. All fragments have access to the *SelectionHelper*, *ImageCollection* and *ImageCollectionPositions*. Furthermore each RecyclerView has a listener (color coded green), which processes the user input and a decoration (color coded violet) to highlight items within the RecyclerView, in this case thumbnails. The decorations are used to indicate group items, used items and selected items in the collection. In this application the fragments bundle the logic concerned with a RecyclerView, and build an interface between their contained elements and the containing activity. for example they invoke methods of the activity, if needed, or vice versa. The RecyclerView and the corresponding components take over the majority processing part.

- **Listener.** Each RecyclerView has it's own RecyclerView.OnItemTouchListener[16], which detects and processes touch input. It differs between valid gestures and invokes the respective method in the implementation of an OnItemClickListener interface, which is defined in the RecyclerView.OnItemTouchListener implementation. For example is a flick gesture on a specific image item in the RecyclerView detected and propagates it to the OnItemClickListener, the method in the OnItemClickListener takes actions respectively. In this example it calls the method in the *ImageCollection*, which moves the image.

- **Decoration.** A RecyclerView.ItemDecoration[17] can be appended to a RecyclerView. It facilitates the drawing under or over items in the RecyclerView. In this application it is used to indicate groups of images (automatic and manual collapsed images in pink and green), selected images and used images.

- **Adapter.** A RecyclerView.Adapter is required for a RecyclerView. It keeps the data, that is supposed to be displayed in the RecyclerView and takes care of providing and showing the data items. Here, the library "Glide"[18] is used for loading image files to display in the RecyclerView grid. The data for the adapter is managed by the *ImageCollection*. If data which also influences other RecyclerViews is altered, a callback is propagated to the respective adapter.

- ***FragmentStateData.*** This class is implemented as a singleton. It keeps track of the fragments, when a user switches between fragments or activities. It saves the current *GalleryTopFragment* and the scroll positions for all fragments.

[16] `https://developer.android.com/reference/android/support/v7/widget/RecyclerView.OnItemTouchListener`, last accessed: 02/19/2019

[17] `https://developer.android.com/reference/android/support/v7/widget/RecyclerView.ItemDecoration`, last accessed: 02/19/2019

[18] `https://bumptech.github.io/glide/`, last accessed: 02/19/2019

- ***SelectionHelper.*** The *SelectionHelper* is also implemented as a singleton. It can be accessed globally and keeps track of currently selected image items in the RecyclerView. It saves the positions of selected images as integers for each RecyclerView, it keeps lists with permanent selected items and lists with temporary selected items as auxiliary variables to process input.

- ***ImageCollection.*** This class, also a singleton, is part of the core of this application. It loads images from a folder into the collection at the start of the application. It reads the dates and puts them into a map. The dates are processed to create *ImagePositionItem*s and *CollapsedImagePositionItem*s, depending on the time difference between the recording dates and times. *ImageCollection* keeps track of the image items. It keeps a list of all image items as *ImagePositionItem*s and *CollapsedImagePositionItem*s. The latter one extends the former one, there is one list for the trash, the collection, and the storyline each. It provides current data lists and also processes all actions, that alter one or more lists. For example, if an image is moved from the collection into the storyline, it updates both affected lists. Furthermore this class takes care of saving copies of the images in the storyline in the created order into an dedicated folder.

- ***ImageCollectionPositions.*** If a user turns on the position mode, this class takes care of the bottom list, instead of the *ImageCollection*, until the position mode is turned off again. Like the *ImageCollection* this class is implemented as a singleton and is globally accessible. The position mode only affects the storyline and adds a placeholder between all images and at the start and the end. *ImageCollectionPositions* transforms the storyline into a list, which contains those placeholders and takes care of images added or moved to the target position. When the position mode is left, the current list will be transformed back into the standard list by removing the placeholders. When the position mode is turned off, the *ImageCollection* is again used instead of the *ImageCollectionPositions*.

- ***ImagePositionItem.*** An *ImagePositionItem* represents an image in the collection. It saves the name of the file as a string and its date. Furthermore it knows if it is used or if it is a placeholder. If items are collapsed manually or automatically *ImagePositionItem*s are summarized into a *CollapsedImagePositionItem*.

- ***CollapsedImagePositionItem.*** Objects of this type contain a group of automatically or manually collapsed images. They consist of a list of *ImagePositionItem*s and process items which are added or removed from the group.

After a description to the components was given, a simplified procedure of the app will be explained in this paragraph. The paragraph does not depict the full scope of the functionality, only a composition of selected use cases. When the app is started, the *GalleryLowerTopFragment* and the *GalleryBotFragment* are created and build the main view, which is also the start screen. After the *ImageCollection* has been initialized, it loads the necessary information from the image files from the camera folder and creates

ImagePositionItems and *CollapsedImagePositionItems*. Those items are managed in lists by the *ImageCollection*. If a user's action forces changes in those lists, they are propagated to the respective method in the class and the class updates the lists accordingly. A user flicks an image item from the collection in the main view towards the top. The RecyclerView.OnItemTouchListener of the *GalleryLowerTopFragment* detects the gesture and propagates it to the RecyclerView.OnItemClickListener, which further propagates it to the according method in the *ImageCollection*. The *ImageCollection* removes the item from the collection list and adds it to the trash list. The *MainActivity* always makes sure, that the adapter of the secondary affected RecyclerView knows, that the dataset has been changed. A double tap on any image is again detected by the respective listener and propagated further, until the flag for the position mode in the *ImageCollection* is set. The queries in the affected classes know to use the *ImageCollectionPositions* in that case and transform the current storyline into a storyline with placeholders. In case a placeholder in the storyline and an image in the collection are tapped successively, the *ImageCollectionPositions* adds the image at the target position and marks the image in the collection as used. Then the user adds a number of images to the storyline and wants to perform the finesort. A scroll with two fingers to the left in the collection is detected in the *GalleryLowerTopFragment*, this triggers the start of the *FinesortActivity*. When the user is happy with the creation, the save button is pressed and a dialog asks the user to confirm saving the storyline. After "yes" is tapped a copy of each image file with a new name will be saved to a folder in the target order. The images can be opened with a gallery app to swipe through the creation.

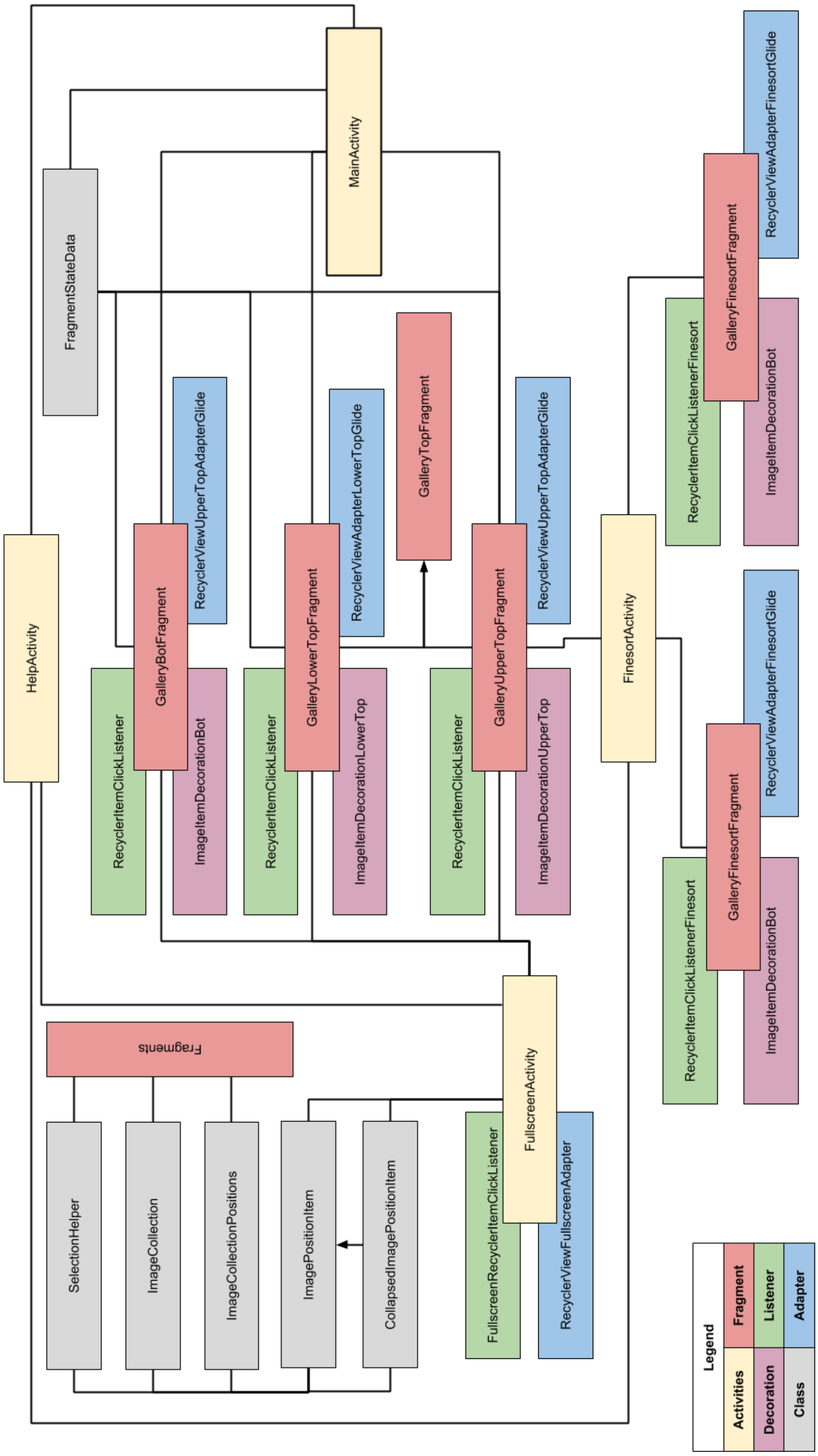


Figure 4.32: Simplified architecture diagram.

CHAPTER 5

Discussion

This chapter reflects on the work done in this thesis. Key points are picked from the UCD process to highlight relevant factors and outcomes. Positive aspects may be helpful for other researchers and weak spots may call attention to potential improvements and suggest future work. Furthermore, the thesis is reflected upon as a whole, raising remarks, which were not mentioned before.

The user-centered design process was started by gathering initial ideas from students to access their understanding of the topic. Their point of view is of special interest as they are not only users of similar technologies in daily life but may already be able to add specific elements relevant for the design of media systems from a “to-be-expert” perspective. Some student assignments revealed creative approaches, which did not fit into well defined categories, like the majority of the other student’s drafts. It was interesting to see, how often two concepts recurred in a similar manner. The first of those concepts was the drag and drop approach, which is common in smartphone applications. The second one was an approach utilizing a splitscreen, whereby the splitscreen served different purposes. Some approaches used one part of the splitscreen as clipboard, some as final result. In comparison to the other student drafts, the splitscreen concept was elaborated more sophisticatedly. The student assignments and the focus group concepts were not fully developed due to limited time. Thereby it should be said that the expenditure of time and effort for a student assignment can not be as extensive as it can be in a thesis. This said, the assignments of students do not examine requirements apart from the given specification and mainly include absolute necessities to fulfill the task. Although this work reflects upon all approaches equally, splitscreen seemed to be somewhat intuitive, drawing on the recurrence of this concept. Furthermore, it is more flexible than some other approaches, which underlines the decision to put more emphasis into this concept.

The participants of the focus group were asked to solve a task like the one in the student assignment. Categories for interface and interaction items and interaction elements, which resulted from student suggestions, were provided as building bricks for the participants to help them solve their task. Creative approaches emerged from the focus group, extending the approaches from the previous step. Since interaction techniques for rearrangement are of a special interest in the thesis, it was interesting to see, which interaction mechanisms were applied in the suggestions. No complex gestures were incorporated in the focus group proposals. Interaction gestures were hardly defined in the concepts, shape or stroke gestures were not involved at all. Therefore the decision was made to maintain this in the draft and not to integrate complex gestures. Using two fingers for multiselection, as it is the case in the prototype, might be at the border to complex gestures and was deemed to be appropriate to handle.

A participant of the focus group stated to use gestures with low effort and not to use gestures like touch and hold or dragging. With the presumption that gestures have to be performed multiple times for a vast amount of images, too much time would be spent on such gestures.

Even though the duration of the focus group was almost three hours, it was not enough time to fully develop an exhaustive draft. Some aspects, like concrete interactions, were mostly neglected. This might have been the case, because it might seem to be unnecessary by the designers to explain them, since some gestures might look obvious within the concept. Many approaches covered grouping of images, and how exactly to accomplish it, but there was little attention paid to the method of how to arrange and rearrange certain images within such groups. Although it was stressed in the introduction of the group phase to think about the order of elements, and not only how to group them, that matter was furthermore neglected in the discussion. However, the focus group raised valuable input for the succeeding work. One topic, which was discussed thoroughly in the focus group, was the treatment of similar images. Some participants posed to take many images of one subject to decide later, which one is the best. They brought up the idea of grouping such images and developed approaches on how to efficiently select the best of these images. Statements like “someone needs to see all images in fullscreen before the decision can be made” heavily influenced the mockup design later on. Some matters in the focus group, like some kinds of grouping or pre-categorization were not pursued, because they were considered to be too complex and too dependent on specific use cases.

In order to process the results from the focus group and the preliminary design direction derived from them, interviews with video editing experts were conducted. While the interviews yielded some valuable information, they also showed that video editing is too specific in some aspects for the topic of this thesis and that its requirements can not just be transformed, to fit into the prototype. For example, the data representation for video clips has to be much more detailed for a professional video editor than for the average user. Video related metadata like the file type or encoding is important in video editing, while it is considered as unimportant in the scope of this work. One participant stated that video editing on mobile phones should be kept simple, which is also a design goal for the developed prototype. In one interview it was mentioned that it is sometimes necessary to try different composition of clips. This also impacts the design of the prototype by enabling quick moving and shifting actions. File organization was stressed to be important in video editing. Although the prototype allows to create collapsed groups, more organizing options might be necessary to fulfill this requirement than the current prototype may be able to provide. Manual groups can not be expanded or edited currently by a user. Furthermore, they are only represented by one image, but if the user is not able to associate the representative image to other images it may be hard to find images or to keep an overview. Titles, descriptions or dates could be helpful to recognize a group of images. Presenting more than one image of a group at once could also help, but their size could be too small to be clearly representative. Dividers or labels between images may also help. There is potential for improvement, but it is also important to find a trade-off, so the app does not become overly complicated by including too complex grouping options.

The evaluation of the prototype led to interesting insights. The final prototype brought up some issues with the automated time-based grouping function. The time frame chosen for this user task was too big, because the subjects could be photographed easily one after each other. Sometimes even 30 seconds were too much and images of different subjects were accidentally collapsed into one group. Therefore it was sometimes hard to find specific images and it was not possible to use more than one image per group, hence the participant had to decide for one image from the group. Also, this led to fewer images and image groups in the collection. Resulting from that, it was not possible to simulate the vast amount of images a user usually has in the image collection, even though the participants were encouraged to take as many pictures as possible. The collapsing function was further discussed by participants, and the comments may lead to possible improvements. Future work should definitely refine automatic grouping to provide more freedom of actions. Manually choosing the parameter for clustering or setting the time frame could improve the usability and the chance to find images. Facilitating the editing of groups is indispensable, for example to disband groups or to extract images from them. All in all the automatic grouping was perceived very well. A way has to be found to make it more accurate and more flexible for the user. Letting a user adjust grouping parameters individually is supposed to help to comprehend, in which group a target image could be found. Resulting from this, a user should have a better overview, by preventing loosing track of images.

The user tests showed that it is important to keep the user updated when changes happen. A user needs to know the current state of the application and in which view a user is currently in. Changes have to be visible. A switch between the views should be clearly communicated. Regarding the views, the differentiation between the finesort and the other views is clear enough, but the difference between rough sort and trash view is harder to grasp. When users cause a change accidentally it should be visible to them and they should be given the opportunity to undo it. Especially the usability tests during the first iteration showed, that participants accidentally moved images to the storyline section, even without noticing their mistake. A possibility to make such events more significant are animations, even though animations may cause the interaction to slow down and break the flow. How to incorporate them may be investigated in future work.

A break in the flow also occurred, because images can not be removed from the storyline in the position mode. This might lead to unintended behavior of the program. Furthermore, many accidental interactions happened in the position mode, when users wanted to open an image in fullscreen but added the image to the storyline instead, because a position was selected.

It could be observed that users rely on known interaction techniques. Even though the interaction techniques were explained to them, people often made gestures they are familiar with, when they forgot how to do something. They often tried to press and hold the finger on the screen and noticed, that nothing happens. It was often mentioned that this gesture could be used for multiselection. Future work could include research about how to implement multiselection in this context using a long tap. Although smartphone users are used to that gesture, it would be interesting, which is more effective in the long-term. Here, the specification did not intend the interaction mechanisms to be already known by users. The interaction mechanisms may be necessary to be learned, before a user can quickly operate the application. The user feedback was positive. Participants highlighted the short response times of the application. They were pleased with the fluent interaction and had no doubt that the handling can be learned and the interaction will get faster over time.

With the incorporation of the position mode, the collapse function was used significantly less often. It was interesting to see how some functions lose some value, because more valuable functions are integrated. Still, it is possible that the specific test task was provoking the preferred use of certain functions over others and the function was therefore less used. Nevertheless, it is considered to be good to have the functions integrated, even if they are used rarely. Participants posed to be able to imagine cases, in which the rarely used functions are helpful, in order to create a customized workflow.

The user-centered approach proved to be fruitful in guiding the design process. This may be illustrated by comparing the initial design concept made up by the author of this work (see Section 4.1) and the chosen functionalities resulting from the design methods approach. The initially drafted concept contained only one view, which was made up by a screen vertically divided into a left and right part. The left part was showing images in a grid, which could then be selected by a "hold and drag" gesture, dropping them in the right part of the screen. In this right part then, the selection was arranged in a grid in the target order. In a group view the right part consists of a scrollable list of groups of images, represented by a row of small preview thumbnails. The "hold and drag" gesture is rather slow compared to a more timely flicking gesture. The importance of low cost gestures in terms of time and effort needed was posed in the focus group discussion, because those gestures were described to be more effective, especially if they have to be performed often. Groups as well as images within groups could be arranged into the favored order. Groups could be named. When opening the group, the images would be displayed in a grid in the right screen-half. In comparison, the images making up the storyline in the implemented prototype are represented in form of a linear stream, similar to timelines in video-editing software. There is also no manual grouping option available in the storyline view, but only in the collection and trash. The workflow of manual grouping works faster and is providing a better overview in the current prototype, as the group's representation is displayed in a less compressed way, compared to the initially drafted concept. Finally, the initially drafted concepts did not contain any mechanism to compare images in a fullscreen view.

Human-computer interaction supported the ordering of images and the approach to the task in three different ways, to answer the research question. First, work in the HCI research area was reviewed for related work, which was previously done by others. Although there was little to be found, which deals with that topic specifically, overlapping areas were surveyed for possible guidance. In this matter image and video browsing, video editing and input modalities with a focus on mobile devices were of interest. Second, methods from the field of HCI were selected to approach the respective topic at hand. The methods were deployed to best fit specific aims throughout the design process. A focus group was conducted to qualitatively investigate creative and valuable approaches and solutions regarding the topic. Interviews led to insights into the needs and perspectives of video editing experts, which can be related to the manual arrangement of media assets in the creation of videos. The results of those methods guided the design of the prototype concept. Throughout the development, user tests were conducted to assure usability and to improve the prototype. The outcomes built a collection of insights, which affected not only the final prototype but also intermediate stages. All results from the applied methods served the user-centered creation of technology in tradition of the field of HCI. A digital, functional prototype was implemented, meeting the requirements as well as nice-to-haves, derived from the methods applied. The outcome was an Android application for rearranging images manually.

How images are ordered on digital devices for image browsing is addressed in the related work chapter. The literature research was extended to browsing and summarization of videos, because they are related to image media. Furthermore, the possibilities of image browsing were explored in a seminar work (see Appendix A), elaborating novel or alternative methods. In most cases images are ordered by their recording date or other metadata. Some projects experiment with the representation of image collections for browsing, but many are not appropriate to be applied on smartphones, because they do not use the limited space of the smartphone display efficiently. Often images are arranged on 3D objects, like a cylinder or a globe. Many concepts from the literature research are based on computations of low-level or semantic feature similarities. Images are then arranged in a grid, a map, on 3D objects or in an orbit around associated images with a high similarity. For video editing, images are arranged on a timeline. Videos are often browsed utilizing a linear depiction of the video. For professional video editing purposes the assets are arranged as thumbnails in a grid. As the work has shown, the requirements of video editing are not convergent to the ones of image sorting.

Direct user manipulation was the central topic of this work. How direct user manipulation can be integrated for rearranging images on mobile phones is coupled to the representation of the images and which options for interaction the representation offers. As it could be seen in the user tests people are used to touch and hold gestures on smartphones, to move items to other positions. In the course of this work low effort gestures turned out to be preferable. Therefore the decision was made to avoid costly touch and hold gestures and consequently also drag and drop gestures. A flick gestures is used to move images from one place to another instead of dragging them over possibly long distances. A mode is included which allows to move images with tap gestures.

Storytelling proved to be an useful example to explain the topic of the thesis. A specific use case was consciously indetermined to provide more flexibility for a user. The work could have benefited from putting emphasis on storytelling as the central theme for the thesis. The prototype might loose universal applicability but might gain functionality specific for that use case. People could easily relate to the storytelling example as main purpose of the final prototype. Encouraging participants to keep that in mind, could have brought insights in this context.

The literature review covered related topics, but to the best of my knowledge no scientific work is concerned with this specific topic. Image and video browsing was presented for different devices, as video editing follows a similar purpose. Video clips are composed, optionally with music, special effects or similar, to tell a story. However, video editing software is far more complex than an app for image rearrangement on a mobile device needs to be. Touch input also limits the interaction possibilities, especially in such a complex context. Furthermore, video media itself is more complex than image media, because videos are inherently time-dependent. Nevertheless, two parts of the application are similar to video editing software and are vital in both applications: the storyline and the collection of assets to work with. The decision to not build up on image retrieval and metadata based algorithms for sorting images was made on purpose, to explore manual arrangement, which apparently receives less attention in existing literature. This work may inspire others to put effort into the topic or to pick up specific portions of it.

CHAPTER 6

Conclusion

This work addressed the manual arrangement of images on mobile phones. It showed in detail how UCD methods were utilized and how requirements for a viable technological solution were derived from them. Their results impacted the development of a prototypical solution by involving potential users but also designers and practitioners. One of the special challenges was to design for a small screen.

A large amount of student assignments, which describe drafts to arrange images on mobile devices were thoroughly reviewed and analyzed. The drafts were dissected into smaller parts and those items were assigned to deduced categories. Items and categories were prepared for a presentation and a folder for a focus group. Participants of the focus group were asked to elaborate their own ideal concept for manual image arrangement. Every participant presented one draft, before they were asked to develop one concept together. Relevant subtopics were discussed and some of them were transformed into ideas for the mockup design. Low-effort gestures are advisable. Alternative image selection is a nice-to-have feature for users who take more images of one subject to have a choice of images later on. Interviews were conducted with video editing experts, because they have to arrange images, video clips or other media assets on a daily basis. One part of the interview was dedicated to the discussion of a mockup, which had all functions intended for the prototype at that time. Feedback from the interviews was incorporated in the concept, which was then implemented as an mobile application. There is a big difference between professional video editing and the aim of this work, therefore only few things could be adapted from the interview. Some ideas from the prototype turned out to be technically infeasible and had to be adapted in the prototype. The development was evaluated iteratively with users, which enabled adjustments based on user feedback in the next iteration.

The final prototype offered the following functionalities: move images, multiselection, collapsing, trash view, position mode and finesort. Even though not all of them have been used in final user tests, they were considered as possibly useful by participants.

Although there is potential for improvement, the users were pleased and the app was described as easy to learn and quick to interact with.

List of Figures

List of Tables

Bibliography

[1] A. Adams and A. L. Cox. Questionnaires, in-depth interviews and focus groups. In *Research Methods for Human-Computer Interaction*, pages 17—-34. Cambridge University Press, 2008.

[2] D. A. Adjeroh and K. C. Nwosu. Multimedia database management-requirements and issues. *IEEE MultiMedia*, 4(3):24–33, Jul 1997.

[3] M. Ajmal, M. H. Ashraf, M. Shakir, Y. Abbas, and F. A. Shah. Video summarization: Techniques and classification. In L. Bolc, R. Tadeusiewicz, L. J. Chmielewski, and K. Wojciechowski, editors, *Computer Vision and Graphics*, pages 1–13, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[4] A. Al-Hajri, G. Miller, M. Fong, and S. S. Fels. Visualization of personal history for video navigation. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, CHI'14, pages 1187–1196, New York, NY, USA, 2014. ACM.

[5] S. Andrews, D. A. Ellis, H. Shaw, and L. Piwek. Beyond self-report: Tools to compare estimated and real-world smartphone use. *PLOS ONE*, 10(10):1–9, Oct. 2015.

[6] A. Aner and J. R. Kender. Video summaries through mosaic-based shot and scene clustering. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision — ECCV 2002*, pages 388–402, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[7] H. Avsar, J. E. Fischer, and T. Rodden. Designing touch screen user interfaces for future flight deck operations. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–9, Sept. 2016.

[8] R. Baecker, A. J. Rosenthal, N. Friedlander, E. Smith, and A. Cohen. A multimedia system for authoring motion pictures. In *Proceedings of the Fourth ACM International Conference on Multimedia*, MULTIMEDIA'96, pages 31–42, New York, NY, USA, 1996. ACM.

[9] L. Barkhuus and V. E. Polichar. Empowerment through seamfulness: Smart phones in everyday life. *Personal Ubiquitous Comput.*, 15(6):629–639, Aug. 2011.

[10] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein. Video tapestries with continuous temporal zoom. *ACM Trans. Graph.*, 29(4):89:1–89:9, July 2010.

[11] F. Bashir, S. Khanvilkar, A. Khokhar, and D. Schonfeld. Multimedia systems: Content-based indexing and retrieval. In W.-K. CHEN, editor, *The Electrical Engineering Handbook*, pages 379–400. Academic Press, Burlington, 2005.

[12] K. Baxter, C. Courage, and K. Caine. *Understanding your Users.* Interactive Technologies. Morgan Kaufmann, Boston, 2nd edition, 2015.

[13] S. C. Bernard. *Documentary storytelling: creative nonfiction on screen.* Focal Press, Amsterdam, Boston, 3rd edition, 2011.

[14] G. A. Boy. *The handbook of human-machine interaction: a human-centered design-approach.* Ashgate, Farnham, 2011.

[15] D. C. A. Bulterman and L. Hardman. Structured multimedia authoring. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(1):89–109, Feb. 2005.

[16] B. Buxton. Human input to computer systems: Theory, techniques and technology - manuscript. `https://www.billbuxton.com/inputManuscript.html`, Jan. 2009. Manuscript of Human Input to Computer Systems: Theory, Techniques and Technology, Chapter 4.

[17] D. Cabral and N. Correia. Video editing with pen-based technology. *Multimedia Tools and Applications*, 76(5):6889–6914, Mar. 2017.

[18] M. G. Christel, M. A. Smith, C. R. Taylor, and D. B. Winkler. Evolving video skims into useful multimedia abstractions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'98, pages 171–178, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[19] R. Compesi and J. Gomez. *Introduction to Video Production.* Taylor and Francis, Oct. 2015.

[20] M. Corporation. Windows touch interactions. `https://docs.microsoft.com/en-us/windows/uwp/design/input/touch-interactions`. Last accessed: 01/21/2019.

[21] K. Dancyger. *The technique of film and video editing: history, theory, and practice.* Focal, Oxford, 4th edition, 2007.

[22] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60, May 2008.

[23] D. Deuff and M. Cosquer. *User-centered agile method.* Wiley, 2013.

[24] G. Developers. Android developers: Activities. `https://developer.android.com/guide/components/activities`. Last accessed: 02/19/2019.

[25] G. Developers. Android developers: Application fundamentals. `https://developer.android.com/guide/components/fundamentals`. Last accessed: 02/19/2019.

[26] G. Developers. Android developers: Fragments. `https://developer.android.com/guide/components/fragments`. Last accessed: 02/19/2019.

[27] U. Flick. *Handbuch qualitative Sozialforschung: Grundlagen, Konzepte, Methoden und Anwendungen.* Beltz, Psychologie-Verl.-Union, Weinheim, 2nd edition, 1995.

[28] M. Fong, A. Al Hajri, G. Miller, and S. Fels. Casual authoring using a video navigation history. In *Proceedings of Graphics Interface 2014*, GI'14, pages 109–114, Toronto, Ont., Canada, Canada, 2014. Canadian Information Processing Society.

[29] I. O. for Standardization. Ergonomics of human-system interaction: Human-centred design for interactive systems. `https://www.iso.org/obp/ui/#iso:std:iso:ts:20282:-2:ed-2:v1:en`. Last accessed: 10/25/2018.

[30] Google. Material design introduction. `https://material.io/design/introduction/`. Last accessed: 01/18/2019.

[31] J. D. Gould and C. Lewis. Designing for usability: Key principles and what designers think. *Commun. ACM*, 28(3):300–311, Mar. 1985.

[32] L. Hardman, Ž. Obrenović, F. Nack, B. Kerhervé, and K. Piersol. Canonical processes of semantically annotated media production. *Multimedia Systems*, 14(6):327–340, Dec 2008.

[33] L. Hardman, J. van Ossenbruggen, K. S. Mullender, L. Rutledge, and D. C. A. Bulterman. Do you have the time? Composition and linking in time-based hypermedia. In *Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia: Returning to Our Diverse Roots: Returning to Our Diverse Roots*, HYPERTEXT'99, pages 189–196, New York, NY, USA, 1999. ACM.

[34] D. Heesch. A survey of browsing models for content based image retrieval. *Multimedia Tools and Applications*, 40(2):261–284, Nov 2008.

[35] P. Hoffmann and M. Herczeg. Hypervideo vs. storytelling integrating narrative intelligence into hypervideo. In *Proceedings of the Third International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, TIDSE'06, pages 37–48, Berlin, Heidelberg, 2006. Springer-Verlag.

[36] L. E. Holmquist. Prototyping: Generating ideas or cargo cult designs? *Interactions*, 12(2):48–54, Mar. 2005.

[37] S. Houde and C. Hill. What do prototypes prototype. *Handbook of HumanComputer Interaction*, 1997.

[38] M. A. Hudelist. Next generation image and video browsing on mobile devices. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, ICMR'13, pages 333–336, New York, NY, USA, 2013. ACM.

[39] M. A. Hudelist, K. Schoeffmann, and L. Boeszoermenyi. Mobile video browsing with a 3d filmstrip. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, ICMR'13, pages 299–300, New York, NY, USA, 2013. ACM.

[40] M. Hurbis-Cherrier. *Voice & vision: a creative approach to narrative film and DV production.* Focal Press, Amsterdam, 2nd edition, 2012.

[41] W. Hürst and D. Darzentas. Quantity versus quality: The role of layout and interaction complexity in thumbnail-based video retrieval interfaces. In *Proceedings of the 2Nd ACM International Conference on Multimedia Retrieval*, ICMR'12, pages 45:1–45:8, New York, NY, USA, 2012. ACM.

[42] W. Hürst, C. G. M. Snoek, W.-J. Spoel, and M. Tomin. Size matters! How thumbnail number, size, and motion influence mobile video retrieval. In *Proceedings of the 17th International Conference on Advances in Multimedia Modeling - Volume Part II*, MMM'11, pages 230–240, Berlin, Heidelberg, 2011. Springer-Verlag.

[43] Ideum. Gestureworks website. `https://gestureworks.com`. Poster of Ideum's Gesturework Framework. Accessed: 07/08/2018.

[44] A. Inc. Apple human interface guidelines. `https://developer.apple.com/design/human-interface-guidelines/ios/user-interaction/gestures/`. Last accessed: 01/21/2019.

[45] J. Jacko. *Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications.* Human Factors and Ergonomics. CRC Press, 3rd edition, 2012.

[46] W. Jackson. *Digital Video Editing Fundamentals.* Apress: Imprint: Apress, Berkeley, CA, 2016.

[47] M. Jago. Benutzeroberfläche von premiere pro kennenlernen. `https://helpx.adobe.com/de/premiere-pro/how-to/overview-interface-premiere-cc.html`, 2018 (accessed January 8, 2019).

[48] Y. Jing, H. Rowley, J. Wang, D. Tsai, C. Rosenberg, and M. Covell. Google image swirl: A large-scale content-based image visualization system. In *Proceedings of the 21st International Conference on World Wide Web*, WWW'12 Companion, pages 539–540, New York, NY, USA, 2012. ACM.

[49] W. Jung and H. R. Yim. Effects of mental model and intrinsic motivation on behavioral intention of smartphone application users. *ETRI Journal*, 38(3):589–598, 2016.

[50] G. Jurca, T. D. Hellman, and F. Maurer. Agile user-centered design. In K. L. Norman and J. Kirakowski, editors, *The Wiley Handbook of Human Computer Interaction*, chapter 6. John Wiley & Sons, Hoboken, New Jersey, 2018.

[51] F. Karray, M. Alemzadeh, J. A. Saleh, and M. N. Arab. Human-computer interaction: Overview on state of the art. *International Journal on Smart Sensing and Intelligent Systems*, 1(1):137–159, Mar. 2008.

[52] T. Karrer, M. Weiss, E. Lee, and J. Borchers. Dragon: A direct manipulation interface for frame-accurate in-scene video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'08, pages 247–250, New York, NY, USA, 2008. ACM.

[53] T. Karrer, M. Wittenhagen, and J. Borchers. Pocketdragon: A direct manipulation video navigation interface for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI'09, pages 47:1–47:3, New York, NY, USA, 2009. ACM.

[54] T. Kindberg, M. Spasojevic, R. Fleck, and A. Sellen. I saw this and thought of you: Some social uses of camera phones. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA'05, pages 1545–1548, New York, NY, USA, 2005. ACM.

[55] Y. Kleiman, J. Lanir, D. Danon, Y. Felberbaum, and D. Cohen-Or. Dynamicmaps: Similarity-based browsing through a massive set of images. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI'15, pages 995–1004, New York, NY, USA, 2015. ACM.

[56] G. Lakshmi Priya and S. Domnic. Modified color layout descriptor for gradual transition detection. In P. Balasubramaniam and R. Uthayakumar, editors, *Mathematical Modelling and Scientific Computation*, pages 421–428, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[57] J. Lazar, J. H. Feng, and H. Hochheiser. Chapter 10 - Usability testing. In J. Lazar, J. H. Feng, and H. Hochheiser, editors, *Research Methods in Human Computer Interaction*, pages 263–298. Morgan Kaufmann, Boston, 2nd edition, 2017.

[58] G. LCC. Material design gestures. https://material.io/design/interaction/gestures.html. Last accessed: 01/21/2019.

[59] F. C. Li, A. Gupta, E. Sanocki, L.-w. He, and Y. Rui. Browsing digital video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'00, pages 169–176, New York, NY, USA, 2000. ACM.

[60] C.-Y. Lin, Y.-C. Chao, and T.-W. Tang. Why not be smarter? examining the factors that influence the behavioral intentions of non-smartphone users. *Industrial Management & Data Systems*, 117(1):32–49, 2017.

[61] T. Low, C. Hentschel, S. Stober, H. Sack, and A. Nürnberger. Visual berrypicking in large image collections. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, NordiCHI'14, pages 1043–1046, New York, NY, USA, 2014. ACM.

[62] I. S. MacKenzie. Input devices and interaction techniques for advanced computing. *Virtual environments and advanced interface design*, pages 437–470, 1995.

[63] T. W. Malone. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst.*, 1(1):99–112, Jan. 1983.

[64] B. Meixner. Hypervideos and interactive multimedia presentations. *ACM Comput. Surv.*, 50(1):9:1–9:34, Mar. 2017.

[65] A. Meschtscherjakov, A. Krischkowsky, K. Neureiter, A. Mirnig, A. Baumgartner, V. Fuchsberger, and M. Tscheligi. Active corners: Collaborative in-car interaction design. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS'16, pages 1136–1147, New York, NY, USA, 2016. ACM.

[66] M. R. Minsky. Manipulating simulated objects with real-world gestures using a force and position sensitive screen. *SIGGRAPH Comput. Graph.*, 18(3):195–203, Jan. 1984.

[67] J. Mota, M. J. Fonseca, D. Gonçalves, and J. A. Jorge. Agrafo: A visual interface for grouping and browsing digital photos. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI'08, pages 494–495, New York, NY, USA, 2008. ACM.

[68] M. J. Muller and S. Kuhn. Participatory design. *Commun. ACM*, 36(6):24–28, June 1993.

[69] C. Nguyen, Y. Niu, and F. Liu. Video summagator: An interface for video summarization and navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'12, pages 647–650, New York, NY, USA, 2012. ACM.

[70] T. A. Ohanian and M. E. Phillips. *Digital filmmaking: the changing art and craft of making motion pictures.* Focal Press, Boston, 2nd edition, 2013.

[71] L. OnePlus Technology (Shenzhen) Co. Oneplus 3t. `https://www.oneplus.com/at/3t`. Last accessed: 11/11/2018.

[72] D. Patel, G. Marsden, M. Jones, and S. Jones. An evaluation of techniques for image searching and browsing on mobile devices. In *Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, SAICSIT'09, pages 60–69, New York, NY, USA, 2009. ACM.

[73] M. Porta. Browsing large collections of images through unconventional visualization techniques. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI'06, pages 440–444, New York, NY, USA, 2006. ACM.

[74] A. Pratt and J. Nunes. *Interactive design: an introduction to the theory and application of user-centered design.* Rockport Publishers, Beverly, Mass., 2012.

[75] J. Preece, Y. Rogers, and H. Sharp. *Interaction design: beyond human-computer interaction.* Wiley, New York, NY, 2002.

[76] O. U. Press. Oxford dictionaries. `https://en.oxforddictionaries.com/`. Last accessed: 03/03/2019.

[77] V. Roth and T. Turner. Bezel swipe: Conflict-free scrolling and multiple selection on mobile touch screen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'09, pages 1523–1526, New York, NY, USA, 2009. ACM.

[78] Y. Rui, T. S. Huang, and S. Mehrotra. Exploring video structure beyond the shots. In *Proceedings. IEEE International Conference on Multimedia Computing and Systems (Cat. No.98TB100241)*, pages 237–240, June 1998.

[79] D. Saffer. *Designing for interaction: creating innovative applications and devices.* Voices that matter. New Riders, Berkeley, Calif, 2nd edition, 2010.

[80] K. Schoeffmann and L. Boeszoermenyi. Image and video browsing with a cylindrical 3d storyboard. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR'11, pages 63:1–63:2, New York, NY, USA, 2011. ACM.

[81] K. Schoeffmann, M. A. Hudelist, and J. Huber. Video interaction tools: A survey of recent work. *ACM Comput. Surv.*, 48(1):14:1–14:34, Sept. 2015.

[82] R. P. Sharma and G. K. Verma. Human computer interaction using hand gesture. *Procedia Computer Science*, 54:721–727, 2015. Eleventh International Conference on Communication Networks, ICCN 2015, August 21-23, 2015, Bangalore, India

Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India.

[83] M. A. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding techniques. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 775–781, June 1997.

[84] B. Still and K. Crane. *Fundamentals of user-centered design: a practical approach*. CRC Press, Boca Raton, FL, 2017.

[85] L. Teodosio and W. Bender. Salient stills. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(1):16–36, Feb. 2005.

[86] L. Terrenghi, T. Fritsche, and A. Butz. Designing environments for collaborative video editing. In *2008 IET 4th International Conference on Intelligent Environments*, pages 1–7, July 2008.

[87] Z. Xiong, R. Radhakrishnan, A. Divakaran, Y. Rui, and T. S. Huang. Chapter 1 - introduction. In Z. Xiong, R. Radhakrishnan, A. Divakaran, Y. Rui, and T. S. Huang, editors, *A Unified Framework for Video Summarization, Browsing and Retrieval*, pages 1–13. Academic Press, Burlington, 2006.

[88] Z. Xu, Z. D. Chen, and H. Nie. Handheld computers: Smartphone-centric wireless applications. *IEEE Microwave Magazine*, 15(2):36–44, Mar. 2014.

[89] R. Yan, A. Natsev, and M. Campbell. An efficient manual image annotation approach based on tagging and browsing. In *Workshop on Multimedia Information Retrieval on The Many Faces of Multimedia Semantics*, MS'07, pages 13–20, New York, NY, USA, 2007. ACM.

[90] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'03, pages 401–408, New York, NY, USA, 2003. ACM.

[91] S. Yim, S. Lee, and S. Choi. Evaluation of motion-based interaction for mobile devices: A case study on image browsing. *Interact. Comput.*, 23(3):268–278, May 2011.

[92] H. Zhang. Introduction to content–based image indexing and retrieval. In K. Jeffay and H. Zhang, editors, *Readings in Multimedia Computing and Networking*, The Morgan Kaufmann Series in Multimedia Information and Systems, pages 247–253. Morgan Kaufmann, San Francisco, 2002.

[93] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. A hand gesture interface device. *SIGCHI Bull.*, 17(SI):189–192, May 1986.

Appendix A

Time-Based Arrangement of Images inspired by Time-Dependent Data Visualizations

Birgit Chmelar
1227330
066 935

188.960 - Seminar on Media Informatics
Seminar Thesis

Advisor:
Dr. Theresia Gschwandtner

Abstract

Images are commonly arranged as thumbnails on a grid. The focus lies on the representation of images, which are ordered in a chronological way. This characteristic is also the case for videos. Video sequences are represented by key frames, which are in video browsing arranged on a timeline. Novel methods for presenting images, which have a specific order are suggested. Those methods are inspired by visualizations of time-dependent data. The suggested visualizations for image browsing are lightweight and build up on thumbnails and metadata instead of dimension reduction methods, which ensures that images are easy to identify and the visualization is easy comprehensible. Examples are given including an overview of the original visualization and the derived mapping between image data and the visualization. The representations have to be complemented by interface interactions like zoom, focus & context, overview & detail to compensate possible loss of space efficiency by the rectangular shape of thumbnails. Characteristics of time and image browsing related themes are reviewed to support the readers comprehension for this topic. Results show that the browser prototypes, especially for mobile devices, still suffer of space efficiency, because time-dependent data visualizations often manifest circular or arbitrary shapes, instead of rectangular which would better fit the rectangular screen. The idea generation for the three deduced image browsing examples was no cumbersome one, which keeps the door open to follow the process with more prototypes. At this point there is no evidence which suggests to resign the idea of adapting time-oriented visualizations for image browsing. However user feedback is deemed as essential for making this decision.

May 17, 2018

1 Introduction

A picture can be expressive as well as impressive. Images are a suited medium to tell or to support stories. Mobile phones offer the possibility to take pictures and videos anywhere and anytime. A reason for recording is often the intention to share experiences and memories with people who have or have not been present at a given event [9] [8]. The more material is recorded, the more important it is to keep an overview and to find the best shots for sharing. For storytelling purposes images have a certain order, which does not necessarily has to correspond with the chronological order in which the images were taken. Image browsing often happens with images arranged on a regular grid [12]. The user is often able to sort them by characteristics, for example metadata, like title, size or date. This work focuses on the representation of images that have a specific order, which can be chronologically or of other nature. In regards of the time factor videos are a good medium to refer to. Videos are a series of images, so called frames. Each frame (except the first and the last one) has a certain predecessor and a certain successor. Therefore the order of the images is crucial and this timebased-media can also be adduced for related work which deals with arranging images in a certain order. Video browsing is as described by Schoeffmann et al. [15] the process to explore a video. Established methods for presenting videos when browsing is by thumbnails of keyframes in a storyboard, which corresponds to a grid, or timeline-based visualization for browsing [5]. Although grids of thumbnails and timelines are a established method for image and video browsing, it can not be known, they are the most effective ones.

Time-oriented data is characterized by the coupling to time [1]. Therefore researchers found and find ways to visualize this kind of data. The link between data and time is vital and can not be neglected to make the exploration possible. Therefore it differs from not time-dependent data and needs different methods of visualization. Aspects of time will be covered in Section 2, Related Work. Data visualizations can serve three goals: *explorative analysis*, *confirmative analysis* and *presentation of analysis results*. Besides aspects of time within visualizations which can be of interest for this work, the purposes will be neglected, the focus lies merely on how time can be visualized and what can be applied to the visualization of series of images. The research field of data visualizations is supposed to inform new ways for this objective.

The aim of the work is to explore the representation of images with a specific order. The specific order can be compared to video, because a video exists of a vast number of images, that have a certain predecessor image and a certain successor image. Video is a time-based media and time-dependent data visualizations will be adduced to inspire novel methods to display images or thumbnails. Concepts will be developed and supported by sketches that explain possible representations and interactions for images and thumbnails. Examples will be approached for desktop as well as for mobile devices. There are no known approaches that use the relationship between time-dependent visualizations and displaying images in a specific order to design representations of images for this purpose. Possible outcomes of this work are the determination that this link between the data cannot be used to create useful representations or to discover a potential in one or more visualizations that emerge from this approach. If it turns out to be the former case it may inform other researchers, about what is estimated as unsuitable. Furthermore it may inspire to investigate off-the-shelf representations for this purpose in unconventional ways.

2 Related Work

The aim of this work is to adduce visualizations of time-dependent data to inform novel methods of arranging thumbnails in a specific order. Although there are no known approaches like this before, there are related of areas with the arrangement of images, like current image and video browsing methods, and time-dependent visualizations which come into play. This section covers shallow topics that are considered important for comprehension.

In file browsers videos are represented by a key frame, therefore the browsing of images and the browsing of videos is similar. Schoeffmann et al. describes video browsing as process to explore a video, whether it is for a directed search, with a clear goal in mind, or to just scan over the content of the video. It makes use of interaction techniques like video navigation or video summarization to facilitate browsing. Storyboards and timeline-based visualizations are common methods for video browsing. Storyboards represent a video in a grid arrangement of thumbnails . There have also been effor ts to arrange storyboards on 3D shapes for videos as well as for image browsing [14]. Another established method for video browsing is based on timelines or timeline-like views. Li et al. [10] developed an advanced browser for digital video by adding some functionalities to classic video browser navigation. A timeline for seeking is named as basic browser control, and is therefore incorporated in common video browser interfaces. Schoeffmann et al. [16] created an video browser which incorporates content analysis. Thumbnails of the images and content-based features are arranged below the viewer window arranged on a timeline. Video Tapestries are an approach with an more abstract use of the timeline [2]. It can not be known, if there are more efficient methods, yet not known, for the visualization of thumbnails with the context of image and video browsing than the established approaches that arrange them on a storyboard or on a timeline.

Apart from grids and timelines more advanced browsers exist, which offer additional or alternative organization by content-based features. This often resides in sorting or clustering by color. Low et al. [11] used color descriptors to create maps and to arrange images based on their color similarity. Barthel et al. [3] adduce semantic image features to determine the similarity between images and arrange them in a map consisting of thumbnails in a grid.

Time-oriented data is specified as data with relevance of time. Time has characteristics which have to be taken into account when dealing with it in the visualization area. Those characteristics can be assigned to categories, which are *scale*, *scope*, *arrangement*, and *viewpoints*. Aigner et al. [1] presents three types of scale, which are shortly described subsequently. *Ordinal* time only represents relative order, it describes time events in relation to other events. The *discrete* scale can additionally also describe distances between time events. Time can be mapped to time units. This time model has its limits in the granularity of the time unit. For example if the smallest unit that can be expressed are hours, no difference can be made between time events which happen in the same hour, if they for example differ in the minutes range. The last type are *continuous* scales. Continuous time models map time to real numbers, it is always possible to integrate further time points between existing points of time. In this work time plays only an abstract role. The aim is to arrange images in a desired order, therefore it is only interesting which images precedes or succeeds another image, which corresponds to a ordinal time scale, however, the relationship of the positions between images does not have to be of a temporal nature. The scope can be classified as *point-based* or *interval-based*. *Point-based* can be imagined as discrete time space, where nothing between time points matters. The counterpart is the interval-based scope, where subsections are of interest. Aigner et al. provide the example of a date, which could exactly refer to the time 00:00:00 or the whole day from 00:00:00 to 23:59:59 in a point based or in a interval-based scope, respectively. The arrangement can be *linear* or *cyclic*, the first one refers to time events which happen one time, the second one to time events which happen repeatedly always at a specific time. The last aspect of time is the viewpoint. The simplest one is *ordered*, where events can be arranged on one timeline, because things happen one after each other. *Branching* can be pictured as a timeline which splits up and each branch depicts alternative sequences. *Multiple perspectives* can express parallel sequences on timelines. Those four aspects give a little introduction into characteristics that account in the modeling of time, they are depicted in Figure 1. It shall be mentioned, that it represents just one snippet of characteristics in the field of time-oriented visualization. This snippet was chosen because it is assumed to be the most relevant one of the topic, covering also the other characteristics would go beyond the scope of this work.

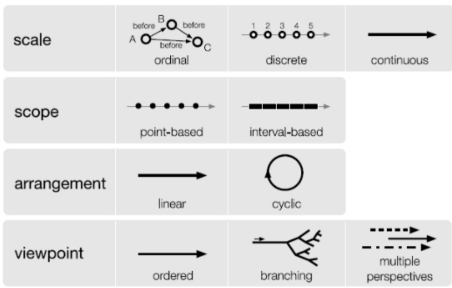
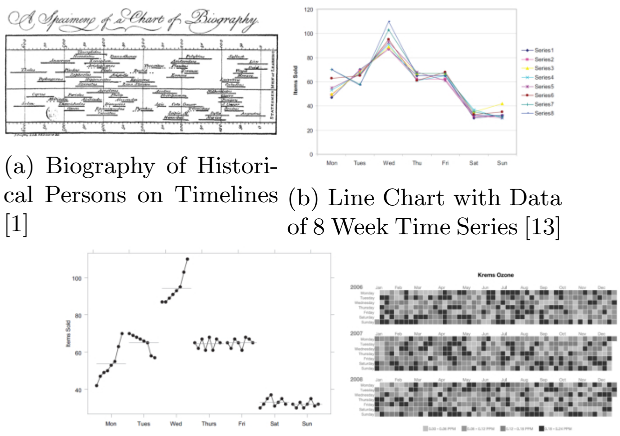


Figure 1: Dimensions of Time [1]

A few examples of basic visualizations of time-oriented data shall be given, before more sophisticated examples will be introduced in the methods part. The represented examples in this chapter are common methods to depict data which depends on time, they are not that powerful like the ones which will follow in Section 3. Figure 2a shows an historical example from 1765 of timelines in data visualization [1]. Joseph Priestley depicts the lifespan of 60 people in one diagram, which was called *Chart of Biography*.

A simple option for visualizing time dependent data are *line charts* [13], where an value of a time series is printed on a diagram, which has often the shape of a day-of-the-week or month-of-the-year. Figure 2b shows an example of a line plot with more than one time series, where each series represents one week. This diagram has the disadvantage of extracting a trend over the weeks. The *cycle plot*, depicted in Figure 2c with the same values, groups the data by each data, which eases the process of comparing the mentioned trend.

Other approaches, which do not use lines, exist. *Tile maps* (Figure 2d) represent values by color- or greyscale-coded intervals, depending on the magnitude of the measured variable [1]. The example shows daily ozone measurements over a period of 3 years, those are classified into one of four intervals. Each tile which represents one day and therefore one measurement expresses one of those four intervals with a specific shade of grey.



(a) Biography of Historical Persons on Timelines [1]

(b) Line Chart with Data of 8 Week Time Series [13]

(c) Cycle Plot with Data of 8 Week Time Series [13]

(d) Tile Map [1]

Figure 2: Basic Methods for Time-Oriented Data Visualizations

The listing of those four examples presents only a small selection of basic visualization methods of time-oriented data. The examples that will be taken for the more practical part of this work will be more sophisticated to leave options for replacing thumbnails or to integrate thumbnails instead of just data points. Next a small introduction into visual storytelling will be given to make the reader more familiar with the topic.

Brehmer et al. [4] approached storytelling from another direction for data visualization. The authors developed a design space for visualizations which is capable to represent narrative points in time. Brehmer et al. reviewed 263 timeline and timeline related research articles to deduce the design space, which considers three dimensions. Those dimensions and the possible elements in each category resulted from the research and were former used in the visualization of timelines. The three dimension spaces are *representation*, *scale* and *layout* and are shown with the extracted possibilities for each in Figure 3. The first dimension refers to the representation of the timeline. The second one, the scale, describes the mapping between temporal distances and the display, indicating characteristics like order, duration and synchronicity between timely events. The third dimension determines relations between groups of events. Each shape of the three dimensions is described in the original paper.

An analysis of each possible combination of the elements was performed, leaving 20 viable combinations for timelines. Figure 4 shows a sample with high diversity of dimensions of those 20 timeline visualizations.

The perspective of Brehmer et al. is rather abstract. The goal in this work is storytelling, which is supported by images or thumbnails, depending on the available space. Aigner et al. [1] devote a few pages to the theme of "Time

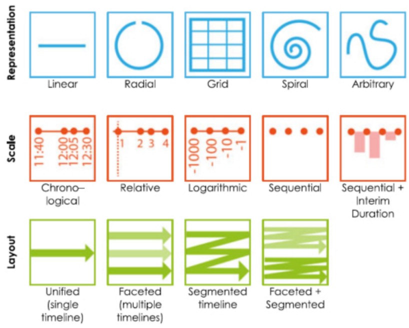


Figure 3: Dimensions and Representations for Time Characteristics used by Brehmer et al. [4]

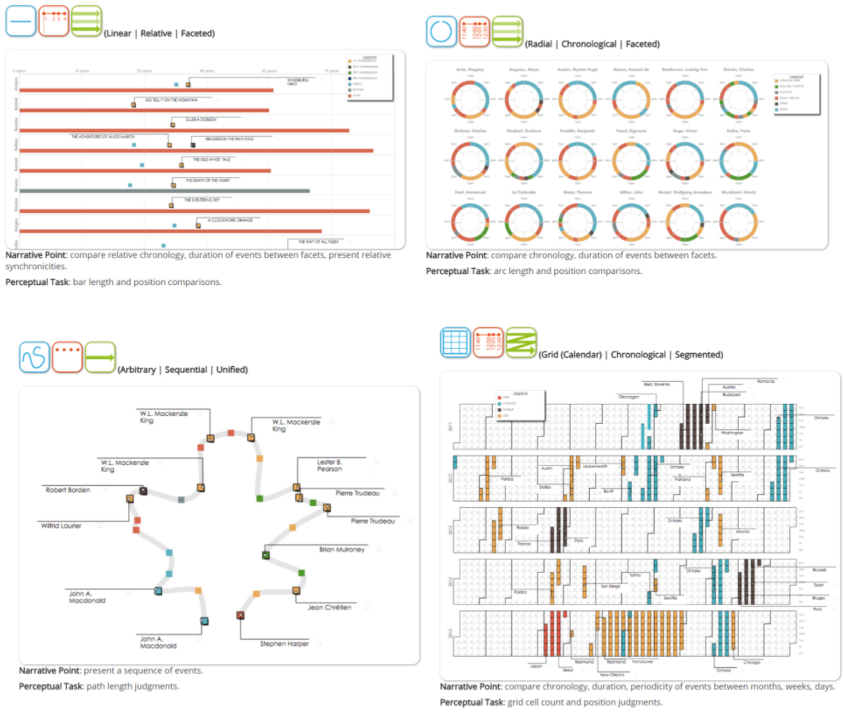


Figure 4: Extraction of Design Space Examples of Brehmer et al. [4]

in Visual Storytelling & Arts". They take user manuals with visual explanations step by step instead of text. Furthermore the authors summarize comics, music notes, movies and paintings in the aforementioned category. Comics are composed of a series of pictures which are located next to each other, which are watched and read in a predefined order. Apart from the time which is contained in the arrangement of pictures each picture does not necessarily show only one moment but can illustrate one scene, which can last longer than one moment. Music notation includes information about durations of notes, breaks, etc. which are contained in symbols. Those examples are supposed to show the possible spectrum of visual storytelling. Considering the arrangement of comic images and neglecting other characteristics, they come close to storytelling in the way that is aimed in this work.

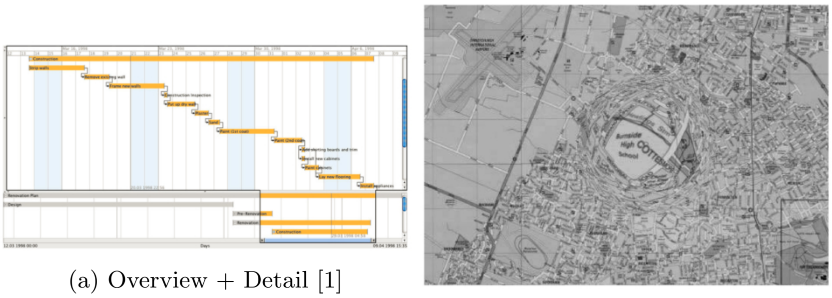
Data visualization systems make use of interaction techniques when dealing with complex data or with an amount of data that does not fit on the screen at the same time. Those interaction mechanisms shall also be covered to introduce the reader into some terms and principles which are later of importance. Cockburn et al. [6] extracts four categories of possible attempts to the lack of space for information a researcher needs to investigate. The categories are classified by different methods to provide a compromise between the amount of data and the loss of overview when navigating within the data. Moving through information by paging, scrolling or panning and spatially arranging data in windows, menus or similar are two options to cope with the mentioned problem. Although this techniques support the user in his task they are only able to cover the problem partly while carrying other disadvantages for the user. Other helping mechanics exist to support the user experience by enabling a smooth transition between rough display of data and a more detailed view. Three of the four by Cockburn et al. reviewed techniques deal with optimizing the exploit of available space, to support data investigation. Those are *Overview + Detail*, which makes use of spatial separation of the overview and the detailed view [1], *Zooming*, which makes use of temporal separation and *Focus + Context*, which provides a focussed perspective within an context view. The last technique reviewed by Cockburn et al. are *Cues*, they support the user by guiding his attention to or away from specific items.

1. **Overview + Detail.** The concept of Overview + Detail is to spatially separate both views. The interfaces are often realized in a way that user interaction in one views impacts the other view in the same way. Additional features ease the navigation in the information space. Cockburn et al. mention the scale ratios of both views, the relative size and position of the detailed view in the overview. A simple realization of overview + detail is a scrollbar, which is often part of user interfaces. The position and the sice of the "thumb" or "knob" indicates the position within the whole document and the scale of the viewed snippet in comparison to the complete size of the document. An example is depicted in Figure 5a. Lenses are also classified as Overview + Detail by Cockburn et al. Although they do not separate both views in an obvious way, they separate the views in a z-direction by blending the detailed view on top of the overview.

2. **Zoom.** The zoom technique provides the ability to switch between different level of details, but in comparison to Overview + Detail there is no separation of views with different level of details. Therefore it is temporarily separated. Only one view which shows a desired level of detail is existent at one time, this view ranges from detail to overview at a time.

3. **Focus + Context.** This technique has, like overview + detail, the ability to show different level of details at one time. In comparison to Overview + Detail does Focus + Context integrate the focus view in the context view. Both are visible at one time and the relation of the position between both views is more clear to the user. One possibility of the realization of Focus + Context is shown in Figure 5b.

4. **Cues.** Cockburn et al. describe that the aim of cue-based techniques are to guide the user, which can be facilitated by highlighting or suppressing specific elements in the visualization. This can, for example, be applied on filtered result sets of items. It can be used in combination with each of the other three aforementioned techniques. Incorporating cues can be achieved by rendering potentially interesting items, like search or filter results, differently than others. The authors name two examples for cue-based techniques: *semantic depth-of-field* methods, which alter the focus of elements, or *decorations*, which call the users attention on items which are originally out of the display space, by rendering some cues on the display edges for their existence.



(a) Overview + Detail [1]

(b) Focus + Context [6]

Figure 5: Examples of Visualization Views

Aigner et al. [1] give an introduction into three key methods for interaction control in visualizations, besides sliders and similar mechanisms in panels or by navigating with buttons. The three methods are called *direct manipulation*, *brushing & linking* and *dynamic queries*.

1. **Direct Manipulation.** This method facilitates the manipulation of the visualization in a direct way. For example zooming by operating the mouse wheel. Manipulating the time axis in time-oriented visualizations in a direct manner is not done by interacting with a slider but with an method, which depends on the visualization itself, therefore it is not possible to describe such manipulations in a standardized way. The example in Figure 6 is supposed to emphasize the specifity of such possibilities. Direct manipulation of the visualization is facilitated by arrows at the beginning and the end of the spiral. This can not be inherited in any visualization. The arrows can be used to navigate in time and by clicking and dragging to manipulate the navigation speed.

2. **Brushing & Linking.** Brushing & Linking works with direct manipulation, its aim is to select elements directly in the visualization. Selection is commonly achieved by pointing and clicking, rubber-band or a lasso tool, which have also the possibility to select more than one element at once. Selected data will be highlighted in all views to show their relation in different perspectives and different information. This is especially useful to select time points or time-intervals to highlight elements, which are related to those times.

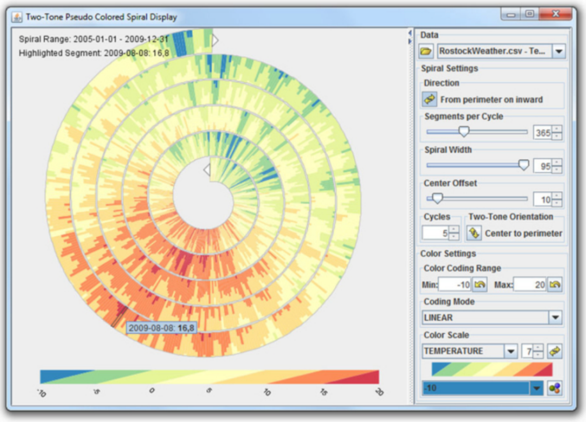


Figure 6: Spiral Visualization with Direct Manipulation [1]

3. **Dynamic Queries.** This technique is often built up on the result of filter queries. Implementations can either highlight the result set of the query or suppress the rest of items that do not fulfill those characteristics. Filter masks often provide the possibility to enter certain values, manipulate them with a slider or filter textually for tags or similar information.

3 Method

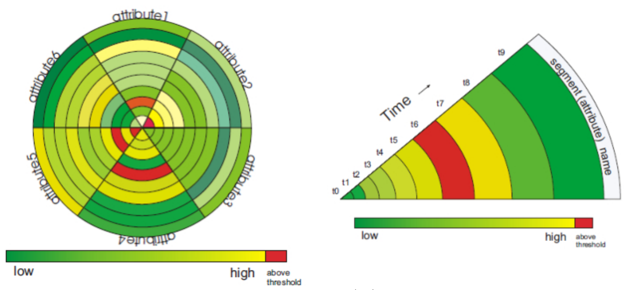
The idea of this work is to inspire browsers for images with a specific order by already existing visualizations of time-dependent data. The aim is not to map the image one-to-one to the known visualization, but to adduce characteristics from the visualization that can be used to show images. The suggested browser prototypes can incorporate interaction techniques like overview+detail, zooming or focus+context [6] to enhance the browsing experience. Cue-based techniques are also of interest in the prototypes. Furthermore this work focuses on the display of thumbnails to make the data easier to grasp and easier to understand instead of using dimension reduction of the images, to project that data onto visualizations in further consequence. More precisely the approach will be as follows: examples will be taken from the TimeViz Browser [18]. The original visualization will be described shortly and the visualization will be taken to inspire a prototype for images browsing. The suggestion for the browser will be described including the mapping from image data to the dimensions, which are originally used in the visualization. Sketches will be used to support the idea of the browser. Pictures have a variety of information, which can be incorporated in the browsing process. Information can be saved in the file as metadata, or it can be extracted by algorithms out of the image, this is possible for semantic as well as for low level features. Teixeira et al. [17] dealt with file searching that takes metadata into account and therefore built a hierarchy of metadata. They extracted the file type, the modification date and the file size as top level and the rest depending on the file type. For pictures they listed *Place Captured*, *Capture Date*, *Kind*, *Camera Brand*, *Camera Model*, *Dimensions*, *Orientation*, *Aperture*, *Flash*, *Focal Length*, *Exposure*. For movie they enumerated *Duration* and *Kind*. Those types of metadata named by Teixeira et al. are candidates for mappings between the images and the visualizations. In this context not every type of metadata will be suitable to use, and since the focus of this work lies on a non-random order, which connects to time, the dates will be of special interest.

4 Results

This section includes the examples that were developed in the course of this thesis. Visualizations from the TimeViz Browser were selected and altered to make them suitable for image browsing. The original visualization is described briefly, followed by a suggestion for an thumbnail mapping. Each subsection represents one example and is named by the original visualization.

4.1 CircleView

CircleView [7] is a visualization for multi-dimensional data with time relevance. It enables the comparison of variables over periods of time. It is supposed to support the user exploring time streams because it eases the grasping of correlations or exceptions. A circle is regularly split up into segments, matching the number of attributes that shall be visualized. Segments are split up into time slots, the duration that corresponds to a slot depends on the data. Figure 7a shows the whole circle, Figure 7b depicts one segment, in which older time slots are further apart of the circle center. This can also be turned around, with most-recent time slots starting at the center and moving towards the edge. Attributes and time are also referred to the independent and the dependent variable, respectively. Areas within the segments depend on the magnitude of the dependent variable.



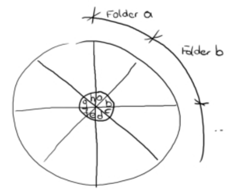
(a) Circle View Complete

(b) Circle View Segment

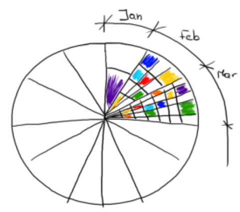
Figure 7: Circle View Original [7]

Adaption. In its original purpose the circle is split up into segments according to attributes and each segment is split up into a number of parts representing the magnitude of the variable for a specific time period. When the Circle View is raised for the purpose of image browsing the idea can be maintained by keeping the time axis along the radial direction. Other attributes of the image can be clustered and mapped to the other dimension of the circle. A simple example is shown in Figure 8a. The circle characteristics are kept of the original visualization to arrange images in this shape. An option is to map the attributes to folders, that were manually created by a user, shown in Figure 8b.

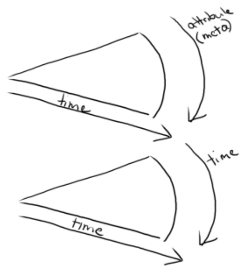
(a) Circle View Image Browsing



(b) Circle View Image Browsing Folder



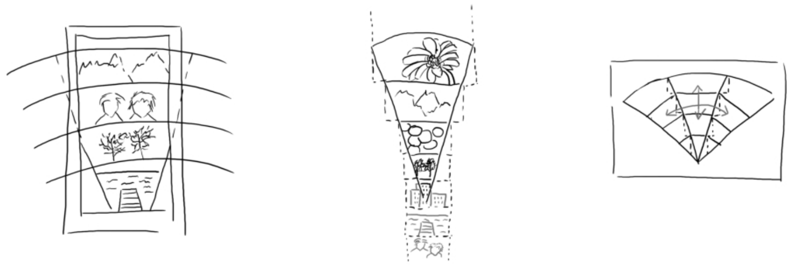
(c) Circle View Image Browsing Months



(d) Circle View Image Browsing Segments

Figure 8: Possibilities of arrangement of thumbnails based on the Circle View data visualization.

Segmenting could also be based on metadata like the place, where it got captured (if available), or the dimension/resolution or the file size. Another possibility is to apply algorithms which either deal with low level content-based features like colors in an image or the extraction of semantic information to cluster images. If time is the only relevant characteristic, then it could also be arranged just in a timely manner. Two concepts of the segment layout are shown in Figure 8d. Segments of the circle could correspond to time periods, as well as the arrangement within an segment. Figure 8c depicts an arrangement based on the months of a year. It shows thumbnails of each image in a segment that is dedicated to a month. The more images in a month were taken the smaller and more densely is the arrangement within a segment. The colors in the image symbolize images, they have no meaning regarding magnitudes. Images in a segment are also arranged in a chronological order. The user would be able to get an overview of how many images he took in which month. The abilities of the browser can be expanded by incorporating zoom, overview+detail or focus+context methods. The incorporation of these methods are essential, because the format and size of thumbnails strongly depend on the size of the display and the amount of images to show. In the case of mobile phones the circle could be rotated left and right and the content of a segment could be browsed by swiping up and down. Figure 9a depicts how the arrangement on a phone could look like. Although only one segment is shown at the time, caused by space limitations, the concept includes the whole circle, but navigation is necessary to browse through the collection. Figure 9b shows a possible scrolling concept along the segment to enable the user to browse through his images although the limited amount of images, again caused by the space limitation. The thumbnails grow bigger the closer they come to the most outer segment. Showing images fullscreen can be incorporated by touching the desired image. Instead of showing the whole circle or just one segment Figure 9c shows an compromise which is suited for landscape on the mobile. The interactions also include rotating the circle and scrolling a segment at a time and zoom in for better view on a desired image.



(a) Circle View Mobile Image Browsing (b) Circle View Interaction Segment (c) Circle View Interaction

Figure 9: Possibilities of arrangement of thumbnails based on the Circle View data visualization.

A disadvantage of using this visualization for image browsing is the shape of the images. By dividing the circle into segments and segments into smaller parts they loose their rectangular shape, which may occlude details of the image that are located near the border or near the edges.

4.2 Intrusion Detection

The visualization *Intrusion Detection* is shown in Figure 10. It is designed for depicting connections from machines and users to a network [1]. The visualization is supposed to support the detection of suspicious user accesses. The time axis is circular and contains 24 hours. Cubes represent machines, accesses from machines are illustrated by straight lines from the machine to the according point on the timeline. On the right image more circles are depicted in a 3D space, each circle refers to one user. In contrast to the hours of a day the visualization is also able to show months or years. Furthermore it is possible to zoom, rotate and filter. Details-on-demand are also available by hovering. Unfortunately there was no meaning of the colors of the machine cubes mentioned in the reference.

Adaption. The original visualization is supposed to ease the detection of suspicious behaviour by supporting the observation of users, machines and accesses including times. Therefore four dimensions are represented in the original system. As before, time is the most important dimension in this work. The circular timeline is kept as well as the option for selecting hours, months or years,

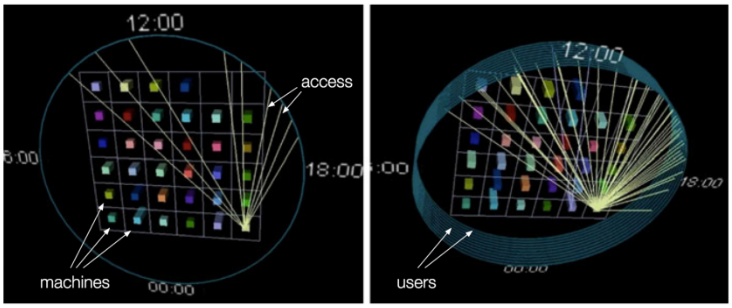
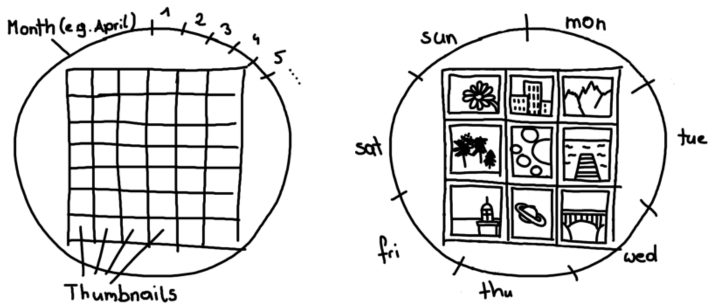


Figure 10: Intrusion Detection Visualization Original [1]

furthermore it is also possible to show days of the week. The basic principle is shown in Figure 11a. The more images were taken in the selected range of time, the more dense is the grid and the smaller the images.



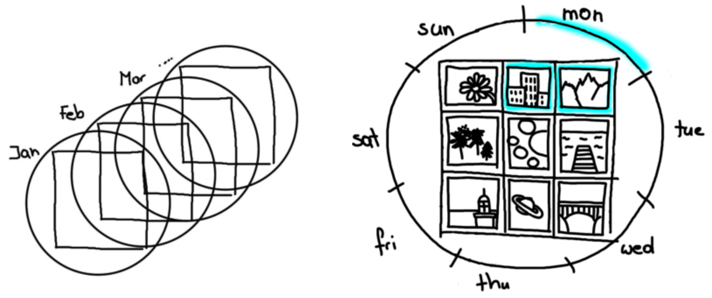
(a) Intrusion Detection Month (b) Intrusion Detection Week

Figure 11: Intrusion Detection Adaptions for Image Browsing

Navigation by mouse, keyboard or touch, depending on the device, shall be incorporated to scroll back and forth through circles, which also alters the displayed images, according to the current time. If hours-of-the-day are selected (Figure 11b), then each ring depicts one day, if days-of-a-week are selected, then each ring depicts one week, etc.

Therefore time in the adapted system can be expressed by two dimensions. The user shall have the freedom to choose and switch between two views: the frontal view and the 3D view. The 3D view similar to the original one is shown in Figure 12a, which can also have smaller distances between the rings if wanted, which was the case in the right picture of the original.

The 3D cubes will be replaced with image thumbnails. There are two options, first, there can be a border left in the matrix between the thumbnails and the grid, which can be later on used for highlighting when time intervals are selected. Or, second, the thumbnails can be placed next to each other without border. One or more intervals of the timeline can be chosen to highlight images, which were taken in the according time frame. The idea of highlighting one interval is depicted in Figure 12b.



(a) 3D Intrusion Detection Months (b) Instrusion Detection Selection Highlight

Figure 12: Intrusion Detection Adaptions User Interaction

In the manner of the original visualization the highlighting could also be achieved by straight lines which connect selected intervals to the images, but this option was rated as unsuitable. The lines would occlude the thumbnails and in most cases the selected images will be next to each other. Pinching gestures on the phone or on touch devices could be used for changing the granularity of the time (e.g. from hours-of-the-day to days-of-the-week) easily step by step. This would rearrange images and change the labels of the circles.

The arrangement of the images on the grid could be chronologically also, or grouped by other characteristics even low-level or semantic features are imaginable.

The advantage of the 3D visualization is that much data can be represented. Apart from the four dimensions in the original (time, users, machines and access) borders of the thumbnails in the matrix can be color coded, referring to groups of characteristics. The disadvantage which is valid for small as well as big screens is the circular shape, which does not match rectangle displays. This is even worse for small devices because the space is limited already and not space-efficiently used with non-rectangular visualizations.

4.3 Anemone

The *Anemone*, Figure 13, shows traffic of users on websites [1]. In this particular case the visitors of the Aestetics & Computation Group at the M.I.T. Media lab are represented. Thick, straight lines in the node-link diagram represent the file structure of the website, labels are optionally available to display names. Node sizes are dynamic and adapt to the number of page visits. They shrink and fade out if pages are visited rarely. Link uses of users within the website are depicted by thin splines of nodes. User interaction enables the selection of nodes to get shown which web page it presents and users have the possibility to drag nodes to move them to another position.

Adaption. The original visualization depicts the static structure of a webpage as a node-link diagram and the frequency of visitors, which impacts the sizes of the nodes. Although the diagram includes two dynamic elements, the

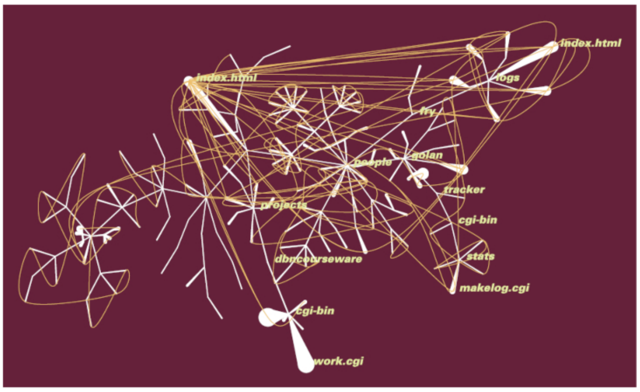


Figure 13: Anemone Visualization Original [1]

size of the page and the splines to depict links, there are few time-dependent aspects in comparison to the visualizations introduced in Subsection 4.1 and 4.2. To adapt the visualization for a time-based arrangement some order has to be introduced that is applicable for images. The suggestion for the thumbnail representation of the node-link diagram is to keep a timeline like a main track, from which branches fork that represent groups of images. The overall arrangement should not be random like in the original to facilitate that images can be found by an user.

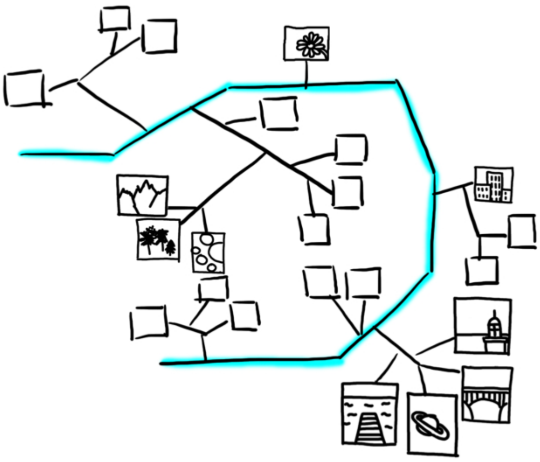


Figure 14: Anemone Visualization Adapted Image Browsing

The idea is shown in Figure 14. The "main timeline" is highlighted in light blue. Groups and subgroups branch from those line to smaller parts of the diagram, thumbnails are located at the end of branches. The granularity of branches, which fork directly on the timeline can be days. Subtrees can have finer divisions, for example hours. Many images which were taken in an time interval can be grouped together. The possibility to adapt the granularity of the timelines are an improvement of user interaction. Like in the original visualization labels shall also be existent, with the possibility to switch them on and off. The classification of the branches within the timeline could also follow folders or structures which were manually created by the user. The arrangement of the images in groups, regarding the angles and length of branches is envisaged as random, but space-efficient, decided by an algorithm. The shape of the main timeline could be more straight or more arbitrary, this decision could be made by the user to support personal preferences. Some possibilites, apart from the more arbitrary example in Figure 14, are shown in Figure 15. Some random parameters could be used in the algorithm to have some variability in the node-link diagrams.

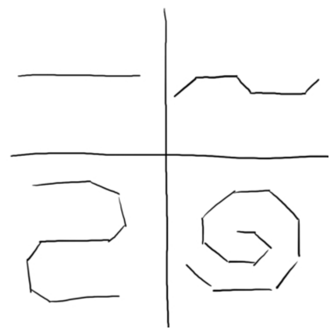


Figure 15: Possible Predefined Shapes of the Timeline for the Adapted Anemone Visualization

Following analogies from the original visualization, images could have a rating of importance, which scales more imporant images to a bigger size dynamically. In simple cases the importance could be based on the number of views of an image of an user, like in the original visualization. Images, which were taken within a short time frame and are therefore summarized in a bigger group, could be more important to the user, assuming that the subject or the subjects were taken more often it could mean that the subject is of special interest for the photographer. An useful mechanism for this image respresentation especially for the use on devices with small displays is zooming. When the user is at a zoom level, at which he is not able to see the whole visualization at once, he can navigate through it like in a model of the universe. The use of a main timeline is supposed to give the user an idea in which direction he has to move, if he is looking for something or if he just wants to browse. For visualizations of many images the application of an overview + detail or focus + context method would be of advantage and to facilitate the jumping to another part of the timeline instead of scrolling to the desired area. Since there are no links in images the curved splines between images additionally to the static lines are considered to be redundant. In an extended version also low-level features or semantic data could be used for the grouping along the timeline.

5 Discussion

The visualizations that were taken from the TimeViz Browser were selected in a way, to achieve a high variability to show diverging possibilities for adaptions. Many visualizations which were or contained inherently a grid or a timeline, were therefore not chosen for this work, even if they were expanded by the display of additional information. After picking one of the visualizations it did not take long to come up with at least one possible mapping of thumbnails, therefore it can be concluded, that there are still more time-dependent data visualizations which can be adduced for the same approach. The attempt was to suggest a possibility that is either feasible for desktop and mobile devices, with bigger and smaller screens, respectively, or to incorporate ideas that ease the interaction on small-screen devices. Although approaches for small screen devices were made, the shape of most visualizations and the need to represent much information is not optimal for smartphones and similar devices. Radial and arbitrary shapes are better suited for desktop devices, which have more space to offer, because the missing space-efficiency is less noticeable. No matter of the device used, interaction mechanisms like overview + detail, zooming and focus + context add a lot of benefit to the image representation. Also cue-based techniques were suggested to improve the user experience. The fact, that only thumbnails and meta-data are adduced for the mapping, instead of low-level features, semantic information or other deduced informations, limits the possibilities of the mapping. As already mentioned this decision was made to ease the fast grasp of the browser systems. The next step would be to evaluate the suggested image browsers with users. Although the visualizations have disadvantages, an essential aspect to decide of their usefulness is feedback of future users. This would provide information if the advantages and the user experience could outperform the disadvantages. Furthermore suggested prototypes could be improved by gaining and incorporating opinions. Although user feedback is deemed as important it was not part of this thesis. In retrospect it would have been advisable to decide for either desktop or mobile devices at the beginning to customize the image browser for one of each type. In consequence the elaborated suggestions are too general and therefore superficial, instead of defining more precisely the incorporation of interaction techniques. It is also noteworthy that the quality of the resulting image browser does strongly depend on the creativity and the ideas of the person, which deduced the suggestion. Therefore it is also possible, that better ways for novel mappings exist for the herein consulted visualization examples.

6 Conclusion

The aim of this work was to adduce suggestions for image browsing systems from time-oriented data visualizations. Visualizations were selected and briefly described from the TimeViz Browser [18], before adapting it for image browsing purposes. The visualization of time-oriented data was chosen as basis because emphasize was laid on an inherent order of the images, which can be imagined as a way to visually support storytelling. The prototypes were described with the help of sketches and some advantages and disadvantages were brought up. The missing space efficiency was deemed as the biggest disadvantage, especially for mobile devices. Interaction mechanisms like overview + detail, zooming or focus + context are therefore considered as vital for the adaption to image browsing. The importance of user feedback for the next step was emphasized and therefore no particular statements were claimed. From the current status and the herein presented work there is no indication to abort the idea of the adaption of time-oriented data visualizations for the mentioned purpose. Although it shall be mentioned that not every visualization is as well suited as another one. User research is recommended for future work and the investigation for alternatives to grids and timelines are still of interest.

References

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Interaction Support*, pages 105–126. Springer London, London, 2011.

[2] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein. Video tapestries with continuous temporal zoom. *ACM Trans. Graph.*, 29(4):89:1–89:9, July 2010.

[3] K. U. Barthel, N. Hezel, and K. Jung. Visually browsing millions of images using image graphs. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, pages 475–479, New York, NY, USA, 2017. ACM.

[4] M. Brehmer, B. Lee, B. Bach, N. H. Riche, and T. Munzner. Timelines revisited: A design space and considerations for expressive storytelling. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2151–2164, Sept 2017.

[5] M. G. Christel. Supporting video library exploratory search: When storyboards are not enough. In *Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval*, CIVR '08, pages 447–456, New York, NY, USA, 2008. ACM.

[6] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, Jan. 2009.

[7] D. A. Keim, J. Schneidewind, and M. Sips. Circleview: A new approach for visualizing time-related multidimensional data sets. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 179–182, New York, NY, USA, 2004. ACM.

[8] D. Kirk, A. Sellen, R. Harper, and K. Wood. Understanding videowork. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 61–70, New York, NY, USA, 2007. ACM.

[9] D. Kirk, A. Sellen, C. Rother, and K. Wood. Understanding photowork. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 761–770, New York, NY, USA, 2006. ACM.

[10] F. C. Li, A. Gupta, E. Sanocki, L.-w. He, and Y. Rui. Browsing digital video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 169–176, New York, NY, USA, 2000. ACM.

[11] T. Low, C. Hentschel, S. Stober, H. Sack, and A. Nürnberger. Visual berrypicking in large image collections. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, NordiCHI '14, pages 1043–1046, New York, NY, USA, 2014. ACM.

[12] M. Porta. Browsing large collections of images through unconventional visualization techniques. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '06, pages 440–444, New York, NY, USA, 2006. ACM.

[13] N. Robbins. Introduction to cycle plots. 05 2008.

[14] K. Schoeffmann, D. Ahlström, and L. Böszörmenyi. 3d storyboards for interactive visual search. In *2012 IEEE International Conference on Multimedia and Expo*, pages 848–853, July 2012.

[15] K. Schoeffmann, M. A. Hudelist, and J. Huber. Video interaction tools: A survey of recent work. *ACM Comput. Surv.*, 48(1):14:1–14:34, Sept. 2015.

[16] K. Schoeffmann, M. Taschwer, and L. Boeszoermenyi. The video explorer: A tool for navigation and searching within a single video based on fast content analysis. In *Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems*, MMSys '10, pages 247–258, New York, NY, USA, 2010. ACM.

[17] P. Teixera, D. Gonçalves, and M. J. Fonseca. Interactive file searching using file metadata and visualization. In *Proceedings of the 24th BCS Interaction Specialist Group Conference*, BCS '10, pages 27–35, Swinton, UK, UK, 2010. British Computer Society.

[18] C. Tominski and W. Aigner. TimeViz Browser. `http://browser.timeviz.net/`, 2015 (accessed April 3, 2018).

Appendix B

Focus Group Guide

Manual Rearrangement of Images

7.8.2018

Introduction

The aim of the thesis is to find effective interaction techniques for manually rearranging images on mobile devices. The work is motivated by storytelling. Images that were taken with a mobile phone sometimes do not match the order, in which the storyteller wants to refer to them. Algorithms are not able to follow the order that is aimed by the storyteller, therefore a method has to be found, which enables the storyteller to arrange images manually. The rearrangement should take place directly on the mobile phone, were the images are stored. Therefore a requirement for the interaction technique is that it is suited for small screen devices with touch input. The interaction is supposed to be effective, which means it should not be time consuming for a high amount of images and the interaction should be convenient for the user. The introduction may need a time for the user to learn and adapt to it.

Procedure

1. Introduction of the topic
2. Introduction of predefined categories
3. Individual elaboration of one suggestion
4. Discussion of the suggestions
5. Cooperative elaboration of one suggestion
6. Discussion of the suggestion

Instructions

The task is to elaborate an approach for small-screen touch devices for a manual arrangement of images. The order of the arrangement is motivated by storytelling, therefore only the storyteller knows, which order matches the story best. An application example: he or she took 100 – 300 images in his or her holiday, but the chronological sequence does not fit the order in which he or she wants to tell the story. Furthermore he or she may not want to use all images, that were taken, but only a subset of them.

A set of categories related to the task are attached in the document, to support the participant. They cover possible solutions regarding the display, the interaction and the system support. The categories were deduced from 140 students with a similar assignment in a written document. The categories and items within the categories are not complete and can be extended. They serve as a pool to provide concepts which can be used to design a system which supports the user in his images sorting task. The concepts may be assembled in any order.

Overview

Display	Representation	Grid Timeline Liste Cover Flow #1 Cover Flow #2 Aero Flip Circle View Gridcircle Anemone
	Views	Fullscreen Fine / Coarse View Clipboard 3D View (if available) Calendar / Date View Old / New Order Overview Current Selection Folders/ Tags / ...
Interaction	Gestures	Gestures & Multitouch Gestures Magic Corner Magic Area Swiping Images
	Selection Method	2 Modes - Single / Multiselection Lassotool Touch &Hold Selection Rectangle Swiping First &Last Image Selection
	Method	Order of Touch Swap Position Arrange Cut &Paste
	Aggregation	Group / Album / Folder Tag Priority Weighting Favorites
System	Sort	by Metdada by Groups by Tags by Priority by Weighting by Favourites
	Filter	

Terms

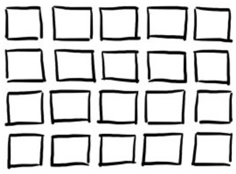
Display

Display describes visual characteristics of the interface, that are considered as important for the purpose of the topic. Here it consists of two subcategories: representation and views.

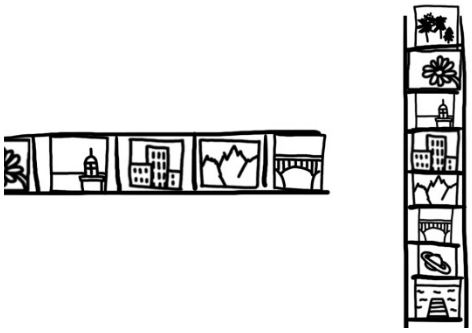
Representation

The term representation is chosen to describe the way, in which a collection of images is displayed on the screen. The most common representations, which appear in literature for image and video browsing, are grids and timelines. Images are displayed as thumbnails, which are small versions of the images.

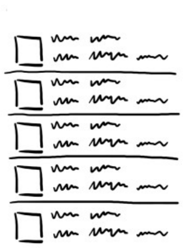
Grid: All images are arranged as thumbnails in a grid.



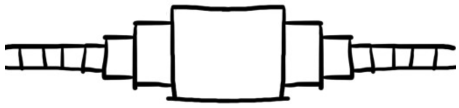
Timeline: Thumbnails are arranged in a sequence. The direction of the timeline is variable.



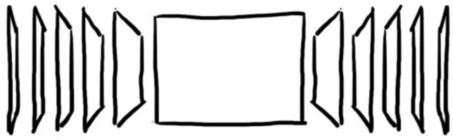
List: Thumbnails are located next to metadata of the according image. Each image is represented by one entry, the entries are arranged one below another.



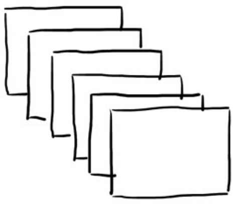
Cover Flow #1: Images are displayed frontally in a sequence, the current image is upscaled, pictures close to the current image have a linear scaling factor, images further apart are only represented in a small version.



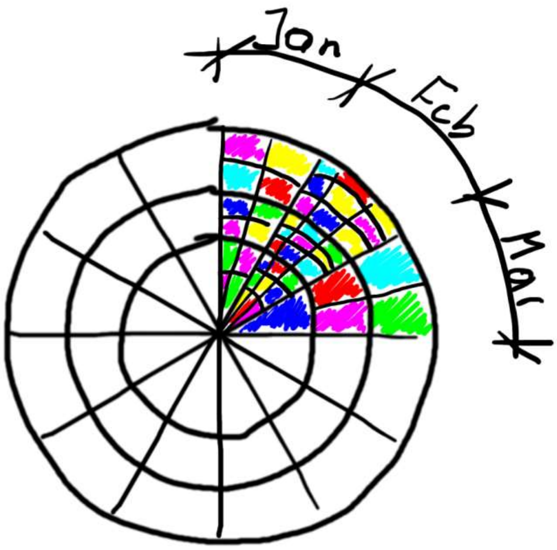
Cover Flow #2: The current picture and pictures close to the current image are linearly scaled in size, growing bigger the closer they are to the current image. In contrast to Cover Flow #1 images apart from the current picture are not arranged frontally but perspective.



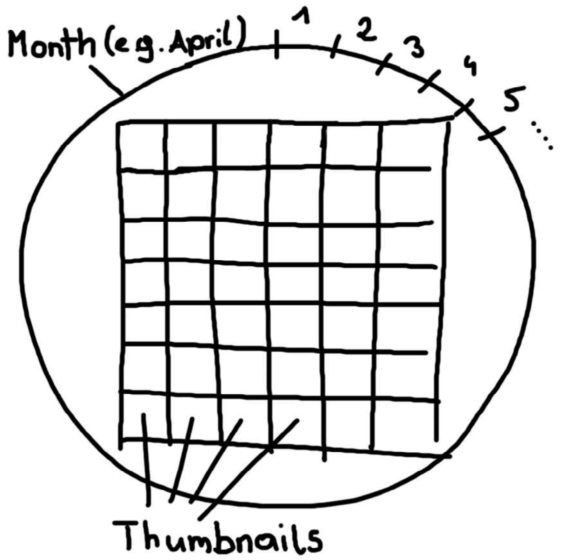
Aero Flip: Images are represented frontally, the active image is in the front, the other images are displaced to the back. The user can scroll through them to change the active image.

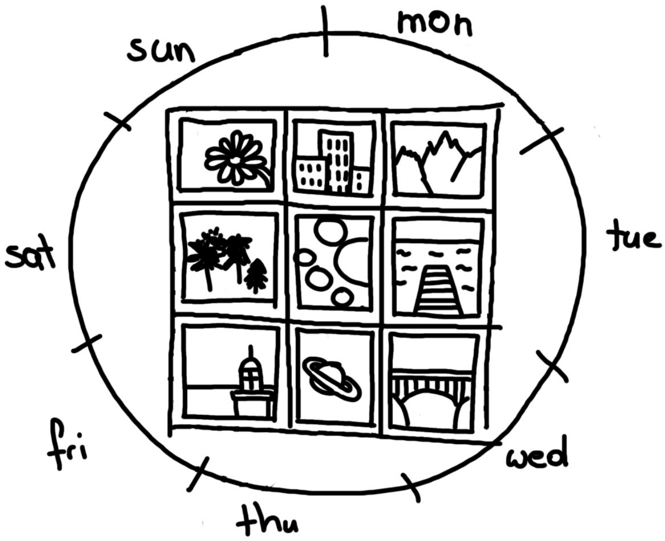


Circle View: Images are arranged within segments of a circle. The user can rotate the circle and zoom to get a closer look of the images. Images lose their rectangle shape. Images can be arranged within the circle by a monthly separation of the circle. It is also possible to view only one segment of the circle and to change the current segment, which is shown.

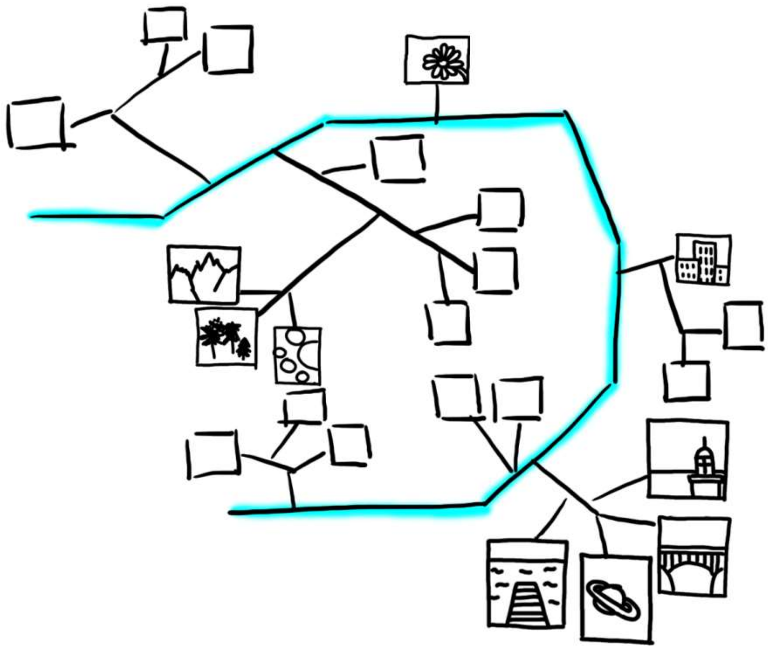


Gridcircle: The images are arranged in a grid, a circle surrounds the grid. The user can interact with the circle to filter periods of time, images which were taken within this period are highlighted. The representation also offers a 3D view, where each month is represented by one grid and circle combination, the user can browse through the months. The more images were taken the denser is the grid.





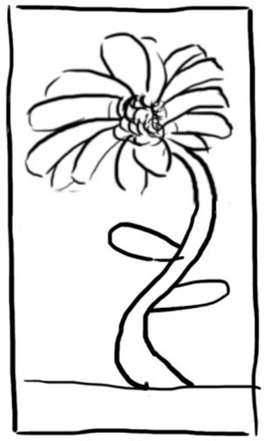
Anemone: Thumbnails are arranged along a timeline. Images are chronologically clustered on branches which split from the main timeline. Images, which were taken with little time periods between are arranged closer together in branches. The user can navigate over the anemone by swiping, like on a map. He can follow the main timeline to have a straight chronology or take shortcuts if he wants to scroll faster back and forth in time.



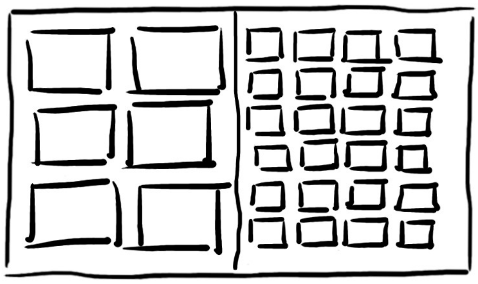
View

A view displays one or more pieces of information, which were categorized as visual characteristic. Views in this sense can be imagined as windows, which are able to show different aspects of the image collection or which provide usable support mechanisms for the user. Those support mechanisms, like the clipboard, add options for the approach of the given task. Representations are also supposed to be displayed within a view. Views can be combined as desired. The size that a view takes on the display can be freely determined. Also besides views that can be shown in parallel on the screen, the interface can be designed in a way in which the user switches between one or more views.

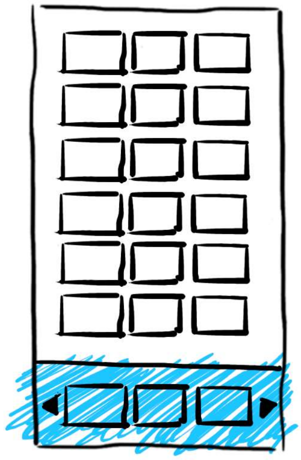
Fullscreen: The user can watch a single image on fullscreen.



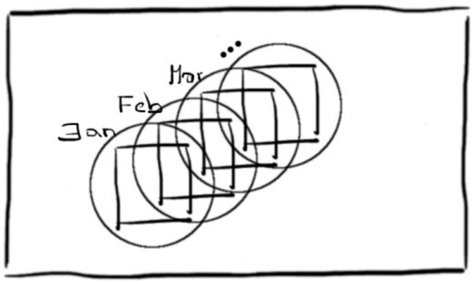
Fine / Coarse View: The user has a fine and a rough view of the images.



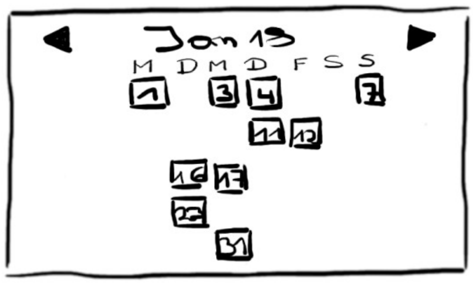
Clipboard: The user has a clipboard, where he can store images temporary.

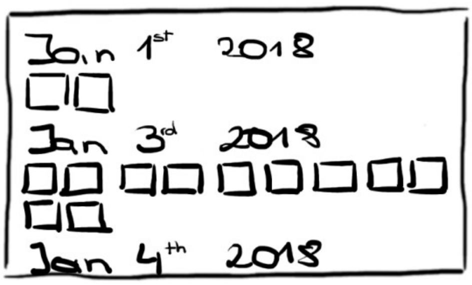


3D View: The user is able to change the view to 3D, if the representation offers the view.

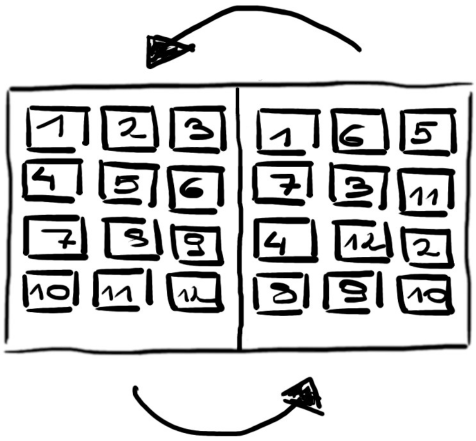


Calendar / Date View: Images are shown with a chronological context which includes dates.

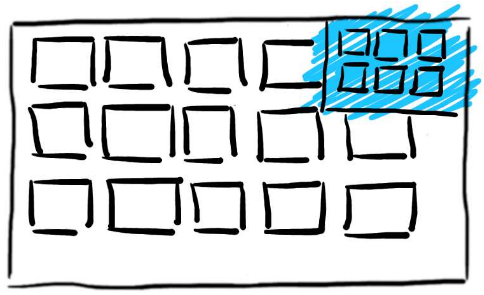




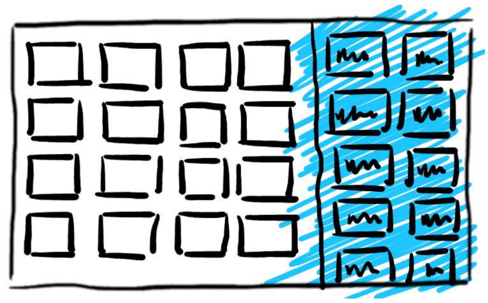
Old / New Order: The user has access not only to the current (new) order of the images, but also to the original order. Either at the same time or by swithing between them.



Overview Current Selection: The interface is able to display an overview of the current selection of the user.



Folders / Tags / …: The user has an overview of his or her created aggregations.



Interaction

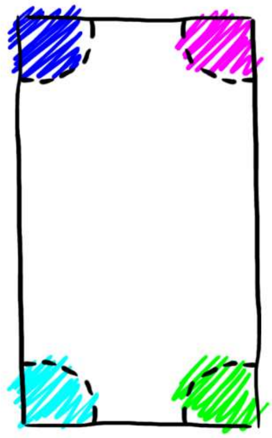
This category describes the way in which the user can interact with the system to achieve the desired result. The interaction happens via the user interface, which consists of the one or more views and one or more representations of the images. The interaction is split up into four subcategories, which describe principles of the interaction and also for the specific task of rearranging images on mobile devices. The four subcategories are as follows: gestures, selection methods, methods and aggregations. The last principal, aggregation, is, on purpose, not elaborated in detail, because they also leave a number of options how to implement those principles. They will be discussed if they turn out to be of interest.

Gestures

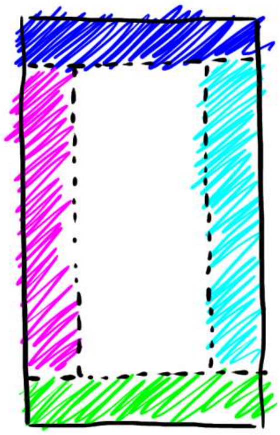
This category includes single and multitouch gestures. They are introduced as a pool of options to interact with the system, they are decoupled of specific purposes. Therefore it is left open which gesture is used to achieve something within the application.

Single & Multitouch Gestures include a collection of gestures. Besides gestures like pinching and swiping the collection consists of gestures taken from gestureworks. The collection can also be extended by individual gestures. Multitouch gestures were taken from gestureworks.com and are located separately.

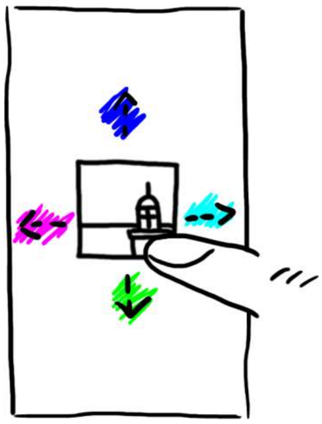
Magic Corners is a concept, in which the user drops an item in one of the corners. Each corner has a different function, the functions can be freely chosen. For example if the user drags and drops one image in the left lower corner it could be deleted from the collection. Dropping the image in the left upper corner could add the image to a group. The advantage of magic corners is that they save space, because they are not visible till an event occurs.



Magic Areas this concept is similar to magic corners. The only difference is that they are not limited to corners but areas in which the user can drop images can be positioned everywhere on the screen.



Swiping Images into a specific direction can have various effects. It is also similar to magic corners and magic areas, with the difference that images do not have be dropped at a specific area, but swiped into a direction. The directions can be assigned to specific functions.

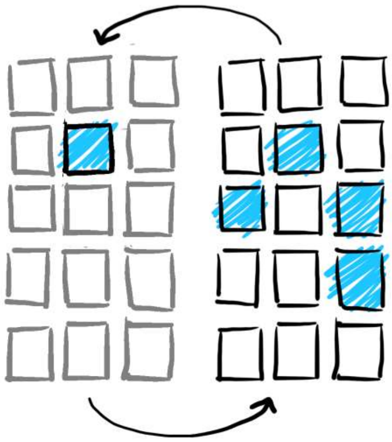


Selection Methods

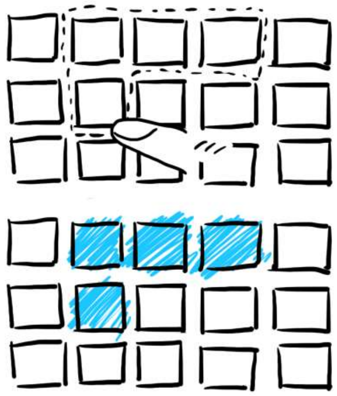
This topic includes options for the selection of multiple images. Since the application is supposed to deal with a high number of images it is of advantage to pick a number of images at once. The content of this subcategory are a collection of suggestions to achieve this.

---

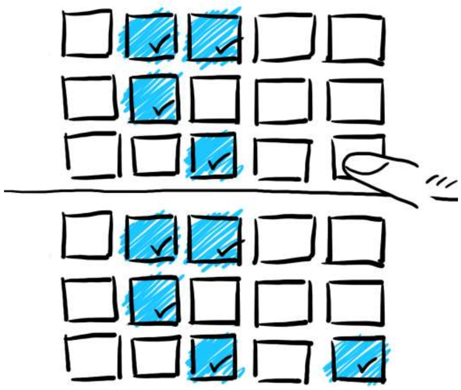
2 Modes - Single / Multiselection: The user can choose between two selection modes, if the selection mode is on. The single selection mode allows him to select a single image with one tap. If the multi selection mode is on he can select any number of images with taps.



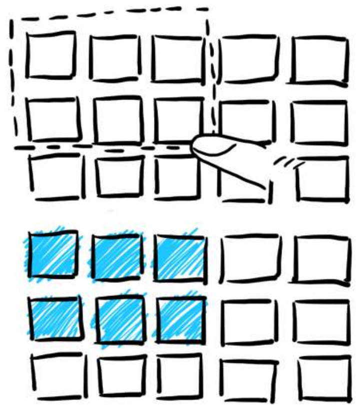
Lassotool: If the user wants to select images with the lassotool he has to swipe an arbitrary, closed shape. All images that are located within the closed shape are selected subsequently.



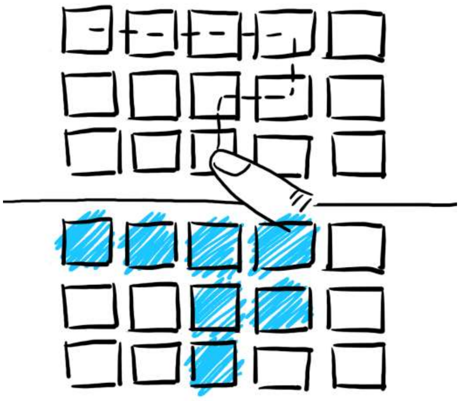
Touch & Hold: The user switches into the selection mode by tapping and holding one image. He is able to choose further images to add to the selection by tapping on them. Holding each image is not necessary if the selection mode is already activated.



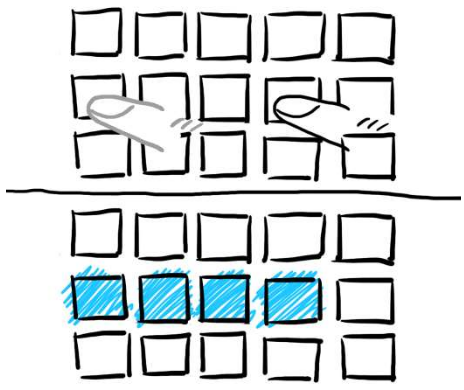
Selection Rectangle: Images are selected by dragging a rectangle on top of the images. All images within the rectangle are selected, similar to the lassotool but less flexible.



Swiping: Swiping over the screen selects all images that were touched.



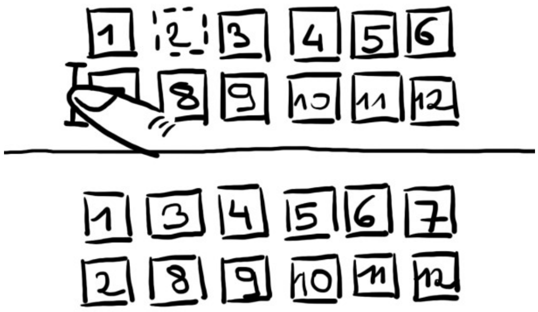
First & Last Image Selection: The user taps on two images, all images in between and the tapped images will be selected.



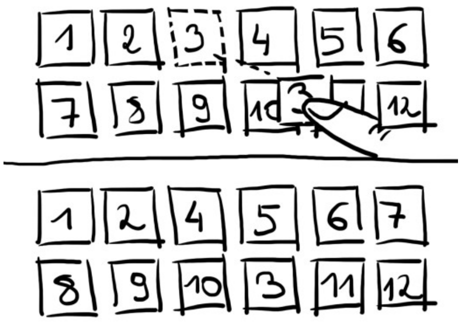
Methods

This category consisits of fundamental approaches for the arrangement and rearrangement of images. Some are inherently coupled to gestures or other interaction aspects, others offer different possibilities for the implementation.

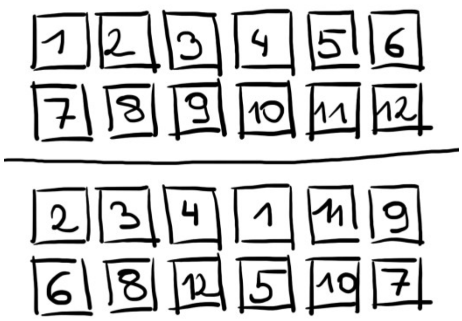
Cut & Paste: The new order is created by cutting images and pasting them before, after or between other images.



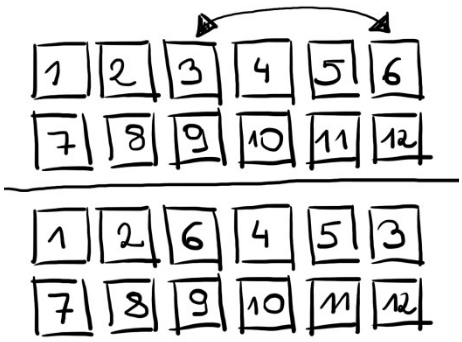
Arrange: The new order is created by moving the images in the representation.



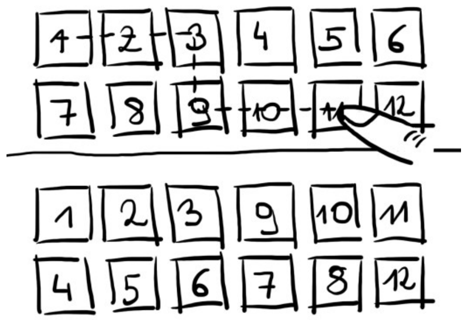
Position: The new order is created by assigning a fixed position number to the images.



Swap: The new order is created by swapping images with each other.



Order of Touch: The new order is created by arranging images according to the order in which images were touched. This can happen by tapping a number of images one after each other or by swiping over images.



Aggregations

Aggregations suggest a number of options for grouping images. The applications should be able to deal with a high amount of images, it is possible that aggregations ease the handling with this amount of images or ease further steps in the process to achieve a specific order of images.

Group / Album / Folder: The user is able to create groups, albums or folders himself. Arranging them in a hierarchical way is possible. Images can be added to those aggregations.

Tag: The user is able to create tags. Those tags can be used to associate images with keywords

Priority: The user can assign priorities to images.

Weighting: The user can assign weights to images. In contrast to priority this leaves more gradations than priority does.

Favorites: The user can mark images as his favorite.

System

This category includes possible features, which are provided by the system to ease the task. The criteria can be something that was defined by the user or retrieved from the metadata of the image. Sort and filter are supposed to support the user to achieve his task, by a systematic pre-selection or arrangement of images. This could be a help if the user is looking for one or more images when he has a rough idea of it.

Sort

Sorting arranges images according to one or more criteria.

Filter

Filtering displays only images which fulfill one or more certain criterias.

Sort / Filter: The user is able to sort or filter by criterias like

- By metadata
- By groups
- By tags
- By priority
- By weight
- By favourites

Appendix C

Interview Guide

1. Für das Zeitnehmen für das Interview bedanken.
2. Consent Form
3. Mich selbst vorstellen, erklären, worum es bei mir geht.

    Medieninformatikstudentin, Diplomarbeit.
    Manuelles Sortieren/Anordnen von Fotos um eine Geschichte zu erzählen und das teilweise als App zu implementieren. Nicht notwendigerweise chronologisch oder nach anderen Metadaten bestimmbar, nur der Autor kennt die beabsichtigte Reihenfolge.
4. Könnten Sie sich bitte kurz vorstellen, wer Sie sind und was sie in Ihrem Beruf tun?
5. Welche Rolle spielt sortieren beziehungsweise ordnen für Sie in ihrem Beruf? Was ist das Ziel davon? Welchen Zweck hat das Ordnen? Gibt es dafür Vorgaben von anderen Personen, dürfen Sie frei entscheiden?
    a. Welche Daten sind nötig (Metadaten)?
    b. Nach welchen Kriterien gehen sie vor?
6. Wie ist ihr Arbeitsablauf? Welche Werkzeuge verwenden Sie dafür?
    a. Was finden Sie gut daran?
    b. Was finden Sie schlecht daran?
7. Wenn Sie sich Ordnen von Fotos am Smartphone in einer App vorstellen, wie würde ihr präferierter Workflow ausschauen. Welchen Zweck verfolgen Sie hier? Ist dieser Zweck anders als der professionelle? Welche Schritte wären Ihrer Meinung nach nötig/sinnvoll/? (Übersicht, Auswahl relevanter Bilder, Verschieben, Suche der Einfügestelle, Fullscreen Bilder für Auswahl,...), Gibt es etwas, das daran besser sein könnte/was nicht funktioniert?
8. Hat die Arbeit im Bereich Ihres Berufes auch Einfluss auf sortieren oder anordnen von Bildern im privaten Bereich?
9. **Vorschlag diskutieren**

    Mockup, imitiert die Funktionsweise. Erklärung dazu, was passiert.
    Ausgangspunkt: oben gesamte Sammlung an Bildern. Die Eckerl unten sagen, dass es Alternativen gibt. Unten ist die neue Anordnung mit ausgewählten Bildern. Ein paar sind unten schon eingefügt. Der grüne Rahmen highlightet Bilder, die schon verwendet wurden.
    a. Mehrfachauswahl
    b. Fullscreen
    c. Collapsen
    d. Moveup
    e. Upper Screen
    f. Positions
    g. Finesort
10. Potenzielle Themen des Mockups
    a. Ordnung/Gruppierung/Vorkategorisierung
    b. Finesort/Nachbearbeitung nötig? Bzw 2 oder mehrphasig? Oder alles auf die gleiche Weise?
    c. Leserichtung
    d. Verhältnis oberer und unterer Teil
    e. Größe Thumbnails?
    f. Querformat/Hochformat
    g. Ausblenden o. Highlighten? (Eckerl vs. Original verändert sich, oder überhaupt auf eine Weise, bei der oben alles gleich bleibt?)
    h. Welche Funktionen ihnen vielleicht abgehen?
        i. Alle Bilder von oben unten einfügen? (Oder mit Vorschlag starten und den verändern?)
        ii. Rückgängig machen?
11. Für das Interview danken.

Appendix D

Einwilligungserklärung zur Erhebung und Verarbeitung personenbezogener Interviewdaten

Interviewerin: Birgit Chmelar, birgit.chmelar@tuwien.ac.at

Dieses Interview wird im Rahmen einer Diplomarbeit eines Medieninformatik Masterstudiums an der Technischen Universität Wien durchgeführt.
Das Interview dient zur Ideenfindung durch die Gewinnung von Einblicken im Bereich der Videoproduktion. Die aus dem Interview gewonnenen Informationen sollen die Konzeption und Entwicklung eines Prototypen für das manuelle Anordnen von Bildern am Smartphone unterstützen.
Die Teilnahme an diesem Interview ist freiwillig und kann auf Wunsch jederzeit abgebrochen werden. Die Daten werden nach Beendigung der Arbeit gelöscht.

Die Daten werden anonymisiert verarbeitet und lassen keinen Rückschluss auf die Person zu.

Die Daten werden nur für die Verarbeitung im Zuge der Diplomarbeit in folgender Form verwendet:

1. Das Festhalten handschriftlicher Notizen.
2. Die Aufzeichnung des Interviews auf Tonträger.
3. Etwaige (partielle) Transkription und Zitation der Tonaufnahme.

Ich, ________________________________, habe obigen Text verstanden und erkläre mich damit einverstanden.

Datum:

Unterschrift (Befragte/r): Unterschrift (Interviewerin):

____________________________ ____________________________

FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Appendix E

User Test Guide

Einleitung

Der User Test besteht aus zwei Teilen: einem Teil, der im Vorhinein vom Teilnehmer/der Teilnehmerin vorbereitet wird, und einem Teil, in dem der Teilnehmer/die Teilnehmerin mit einem Smartphone interagiert, um eine Anwendung zu testen. Im ersten Teil wird der Teilnehmer/die Teilnehmerin gebeten, eine Reihe von Fotos aufzunehmen oder herunterzuladen, um diese digital (z.B. auf einem USB Stick) zum zweiten Teil mitzunehmen. Alternativ zum Herunterladen können die Bilder auch mit einem Smartphone oder einer Kamera vom Monitor abfotografiert werden. Der zweite Teil besteht aus dem eigentlichen Test der entwickelten Software. Dabei wird der Benutzer/die Benutzerin eine Aufgabe mit einer App lösen. Die App wurde im Zuge der Diplomarbeit entwickelt, sie ist dafür gedacht, eine Reihe von Bildern manuell in der vom Benutzer/der Benutzerin gewünschten Reihenfolge anzuordnen. Die Testperson wird gebeten vor Ort eine Einverständniserklärung zu unterschreiben, um der Teilnahme an dem Test zuzustimmen. Im Anschluss wird die Anwendung verwendet, um die Aufgabe zu lösen, begleitend dazu werden während und nach dem Test Fragen dazu gestellt. Die Bilder verbleiben im Besitz des Urhebers/der Urheberin und werden nicht in der Diplomarbeit zu sehen sein. Sie werden nach jeder User Test Sitzung wieder gelöscht.

Aufgabe

Lies die Geschichte auf der nächsten Seite. Hilf Marie, sich noch lange an ihren Tag zu erinnern, indem Du Bilder machst und/oder herunterlädst, die mit den Ereignissen des Tages zusammenhängen. Die App unterstützt den Benutzer/die Benutzerin Bilder in seiner/ihrer gewünschten Reihenfolge anzuordnen und dadurch die sogenannte „Storyline“ zu kreieren. Dabei kann der Benutzer/die Benutzerin entscheiden, welche Motive er verwenden will, je nachdem, was ihm/ihr wichtig erscheint (z.B. ein Foto von einer Müslischüssel, für das Frühstück, ein Bild von einer Krone, für den Rechenkaiser, ein Bild von einem Würfel, für das Ergebnis 6, ein Bild von einem Hund,...) Die Bilder müssen nicht in der gleichen Reihenfolge aufgenommen werden, wie sie in der Geschichte vorkommen. Außerdem ist es erwünscht, dass mehrere Fotos vom gleichen Motiv als Alternativen gemacht werden.
Die Auswahl der tatsächlich verwendeten Bilder wird vor Ort gemacht werden, deshalb wird darum gebeten, möglichst viele Bilder aufzunehmen und mitzubringen, um eine große Auswahl zur Verfügung zu haben. Nicht jedes Ereignis von Marie muss mit einem Bild dargestellt werden, es ist dem Teilnehmer überlassen, welche Ereignisse er mit einem Bild darstellen möchte. Ein Richtwert für die Anzahl an tatsächlich verwendeten Bildern (in der “Storyline”) liegt zwischen 10 und 20 Bildern und einer beliebigen, möglichst großen Anzahl an Bildern in der Auswahl.
Neben frei wählbaren Motiven, die der Teilnehmer/die Teilnehmerin für die Geschichte verwenden möchte, sollten zusätzlich Bilder von folgenden Objekten vorhanden sein:

*Brunnen	*Fahrrad	*Restaurant/Café
*Schuhe	*Zeitung/Prospekt	*Polster

Ein Foto der Glückssocken und ein Foto der Ballontiere werden zur Verfügung gestellt und müssen nicht gemacht werden.

Die Anwendung unterstützt derzeit nur .jpg! Daher die Bilder bitte in diesem Format mitnehmen.

Geschichte

Marie sammelt leidenschaftlich gerne. Sie hat annehmbare Sammlungen von Tassen, Glasflaschen und Hauben, jedoch bestehen nicht alle ihrer Sammlungen aus Dingen einer Sorte. Die Objekte in ihrer Sammlung haben nicht immer eine offensichtliche Gemeinsamkeit. Sie hat eine besondere Art, die Dinge zu betrachten und nur wenige Leute sind in der Lage diese Gemeinsamkeiten in manchen Sammlungen zu erkennen. Eine ihrer Lieblingssammlungen besteht aus Dingen, die lustige Schatten werfen.
Vor kurzem war Marie zu einer Geburtstagsfeier eingeladen und sie hofft, dass sie sich noch lange an diesen Tag erinnern kann. Nicht nur die Geburtstagsfeier, sondern auch der restliche Tag hat ihr viel Spaß gemacht. In der Früh ist sie bereits kurz vor ihrem Wecker aufgewacht, weil sie schon aufgeregt war. Sie aß ihr Frühstück, putzte sich die Zähne und zog sich um. An diesem Tag trug sie ihre Glückssocken. Sie machte sich mit einem Lächeln im Gesicht auf den Weg zur Schule. Es war Freitag, und freitags hatte sie ihr Lieblingsfach: Mathe. Der Unterricht startet jedes mal mit einer Runde des Spiels „Rechenkaiser“. Multiplikationen und Divisionen mussten ausgerechnet werden. Manchmal stellt sich Marie gerne Dinge oder Formen vor, die sie an das Ergebnis ihrer Rechnungen erinnern. Eine ihrer Rechnungen war 48/8 und sie stellte sich einen sechsseitigen Würfel vor. An diesem Tag hat sie das Spiel Rechenkaiser gewonnen, sie war sehr stolz auf sich. Nach der Essenspause hatte sie wieder Unterricht. Am Heimweg begegnete sie einem flauschigen Hund. Sie setzte sich an ihre Hausaufgaben und begann damit, das Datum, der 4. 1., auf ein Stück Papier zu schreiben. Sie musste dabei an ihr Lieblingsbuch und ihren Lieblingsstift denken, weil diese den Schatten eines Vierecks und eine lange Linie, wie eine 1, werfen. Weil sie sehr schnell mit ihren Hausaufgaben fertig war, durfte sie im Anschluss noch fernsehen. Später wurde sie von ihrem Vater mit dem Auto zur Geburtstagsfeier ihrer Freundin gefahren. Marie freute sich schon sehr darauf, ihr Geschenk zu überreichen. Es war eine Tasse aus ihrer Sammlung, ihre Freundin hat letztens erwähnt, wie gut ihr die Tasse gefiele. Marie mochte die bunte Dekoration auf der Geburtstagsfeier, besonders die Luftballontiere. Sie spielten Spiele und aßen Torte und die Zeit verging viel zu schnell, bis sie wieder von ihrem Vater abgeholt wurde.

Appendix F

Einwilligungserklärung zur Erhebung und Verarbeitung personenbezogener Daten im Zuge eines User Tests

Interviewerin: Birgit Chmelar, e1227330@student.tuwien.ac.at

Dieser User Test wird im Rahmen einer Diplomarbeit eines Medieninformatik Masterstudiums an der Technischen Universität Wien durchgeführt.
Der User Test dient der Evaluation des aktuellen Prototypen. Qualitatives Feedback soll eingeholt werden, um etwaige Adaptierungen vorzunehmen oder das Feedback schriftlich in der Diplomarbeit festzuhalten.
Die Teilnahme an diesem Interview ist freiwillig und kann auf Wunsch jederzeit abgebrochen werden. Die Daten werden nach Beendigung der Arbeit gelöscht.
Die Daten werden anonymisiert verarbeitet und lassen keinen Rückschluss auf die Person zu.

Die Daten werden nur für die Verarbeitung im Zuge der Diplomarbeit in folgender Form verwendet:

1. Das Festhalten handschriftlicher Notizen.
2. Die Aufzeichnung des User Tests auf Tonträger.
3. Etwaige (partielle) Transkription und Zitation der Tonaufnahme.
4. Das temporäre Speichern von Bildern beziehungsweise Fotos des Teilnehmers auf dem Testgerät.

Ich, ________________________________, habe obigen Text verstanden und erkläre mich damit einverstanden.

Datum:

Unterschrift (Teilnehmer/in): Unterschrift (Interviewerin):

____________________________ ____________________________

FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics