Domesticating Gestures:
Novel Gesture Interfaces on
Mobile Phones and Tablet Computers

Student: Job van der Zwan
Studies: Interaction Design
Semester: 4
Student ID: 831217T310
Birth date: 17 December 1983
Address: Lönngatan 52D, 1102
Phone-No.: +46762073048
E-Mail: j.l.vanderzwan@gmail.com

Malmö, 19 May 2015
Acknowledgements

I wish to express my thanks to the many people without whom I would not have been able to write this thesis.

First, I would like to thank Crunchfish for the opportunity to do experimental design research around their technologies, and a wonderful internship during and after the thesis project. I had a great time, and am grateful for the privilege of being part of your team for a while.

My thesis supervisor Jonas Löwgren, who set up the internship, provided guidance, and gave impeccable feedback at what often appeared to be superhuman speeds. You are an inspiration, and I aspire to one day be as good at this craft as you.

My examiners Jörn Messeter for his constructive feedback, and Simon Niedenthal for re-examining the final version thesis after I spent some time improving it.

My friend and colleague at Malmö University Jaffar Salih, for graciously spending some of his free time on drawing illustrations to clarify some of the more abstract concepts in this paper.

My classmates at the Malmö Interaction Design Master programme for two amazing years in and outside of Malmö University.

Studying in Sweden would not have been possible without a scholarship of the VSB Foundation, who only asked that I will use the expertise and experience acquired abroad to contribute to society. I am grateful for their faith in me, and will do my best to live up to their expectations.

I would like to thank my loving family for unconditionally supporting me throughout my studies, despite taking so long to find my way. Thank you for your patience, I have certainly tested it.

Finally, I would like to thank my girlfriend, Karolina Vainilkaite, without whom I never would have discovered Interaction Design or thought of coming to Malmö. Thank you for all your support, your love, and for sharing your life with me.
Table of Contents

1 INTRODUCTION AND BACKGROUND.................................................................1

2 THE RESEARCH FOCUS......................................................................................2

3 THEORY, RELATED LITERATURE AND PREVIOUS WORK.............................4
  3.1 Gesture Interface Research............................................................................4
  3.1.1 Classification of different gesture styles.......................................................4
  3.2 Bimanual Input...............................................................................................5
  3.2.1 The Kinematic Chain model........................................................................7
  3.3 A link between design and (post)phenomenology........................................8
  3.3.1 Mediation & Embodiment............................................................................8
  3.3.2 Technical Intentionality............................................................................11
  3.3.3 What postphenomenology implies regarding “natural” user interfaces........14
  3.4 Theory – Summary......................................................................................16

4 METHODOLOGY..............................................................................................17

5 PORTFOLIO – THE EXPLORATORY PROCESS............................................20
  5.1 Week 1 – Opening up the design space..........................................................20
  5.1.1 The smart phone as a case-study of implicit design assumptions..............21
  5.1.2 User scenario: social exploration in the context of grandparents and their grandchildren 23
  5.1.3 Design concept: the tablet as a mobile knowledge database......................24
  5.1.4 Design proposal: the smart phone as a wearable accessory to augment play 26
  5.1.4.1 The Minotaur’s Tail..............................................................................26
  5.1.4.2 The Blinded Cyclops............................................................................28
  5.1.4.3 A futuristic heads-up display..................................................................29
  5.2 Week 2 – Investigating Collaborative Interfaces............................................30
  5.2.1 Crisis Management Environments..............................................................31
  5.2.2 A Collaborative Remote Whiteboard........................................................32
  5.3 Week 3 to 5 – Virtually Extending the Screen..............................................33
  5.3.1 Design proposal: Grab & Drop clipboard..................................................34
  5.3.2 Design Proposal: A full-featured drawing application on a small touch screen 45
  5.3.2.1 Brushes and brush tools........................................................................51
  5.3.2.2 Colour panel..........................................................................................53
  5.3.2.3 Layers, layer options, transformations and filters....................................56

6 THE EMERGING DESIGN SPACE OF GESTURE INTERFACES.....................58
  6.1 Gestures in the context of designing bimanual interfaces..............................58
  6.2 Gestures in the context of social interaction and attention............................58
  6.3 Gestures in the context of embodiment........................................................59
  6.4 Gestures in the context of human ex-centricity..............................................60
List of Figures

Fig. 1: Top: parallel action. Middle: orthogonal action. Bottom: cooperative bimanual action.......................................................................................................................................... 6
Fig. 2: Writing as a cooperative bimanual action................................................................. 7
Fig. 3: Mediated Communication.......................................................................................... 9
Fig. 4: Mediation and Embodiment........................................................................................ 10
Fig. 5: “Put That There,” one of the first multi-modal interfaces featuring deictic gesture input................................................................................................................................. 15
Fig. 6: “The optics form the programmatic frame, the engagements challenge this frame and make it drift, and the takeaways are static snapshots of the current state of the overall program.” – M Hobyе................................................................................................. 18
Fig. 7: The touch-screen mimicking a game-controller.......................................................... 21
Fig. 8: Bodystormed mock-up of the Minotaur Game, using a paper bag and default camera app................................................................................................................................. 26
Fig. 9: The Blinded Cyclops – the phone as an artificial extension of the senses.............. 27
Fig. 10: The Cyclops has seen you!.................................................................................... 28
Fig. 11: A suggested low-fi robot costume.......................................................................... 29
Fig. 12 top: the original image, bottom: a crude mock-up of a fake HUD.......................... 30
Fig. 13: “Towards Seamless Collaboration Media: From TeamWorkstation to Clearboard”............................................................................................................................... 32
Fig. 14a: An application supporting Drag & Drop input. 14b: User opens palm to activate multi-clipboard................................................................................................................................. 35
Fig. 15a: Grab & Drop overlay appears. 15b: Previewing a copied piece of rich text…..... 36
Fig. 16a: Navigating Grab & Drop Interface. 16b: Previewing a copied image................. 37
Fig. 17a: Grabbing and holding copied text. 17b: Dropping held text in an application................................................................................................................................. 39
Fig. 18a: Moving to copied image. 18b: Grabbing previewed image................................. 40
Fig. 19a: Holding copied image. 19b: Dropping held image in an application................. 41
Fig. 20a: Putting down the hand hides the overlay. 20b: Normal interaction with the program resumes................................................................................................................................. 42
Fig. 21: Drawing App mock-up – “brushes and brush tools” menu group............................ 49
Fig. 22: Pulling out a detailed settings menu....................................................................... 52
Fig. 23: A very rough sketch of pulling out the detailed brush settings menu – creating a specific layout for all possible settings was skipped for this mock-up.......................... 52
Fig. 24: “Colour panel” menu group.................................................................................. 53
Fig. 25: Switching brush shape and colour without leaving “peek mode”............................ 54
Fig. 26: “Layers, layer options, transformations and filters” menu group......................... 56
Fig. 27a: Pull-menu showing blend modes. 27b: Inactive ui elements should be hidden to maximise visibility of interaction consequences................................................................. 57
1 Introduction and Background

Gesture based interfaces have been researched for over four decades, but up until recently have always been limited to academic settings. This is because they required prohibitively expensive and large set-ups, usually in the form of an entire room designed specifically around the gesture detection. The introduction of smart phones and tablets has changed this. These devices combine into one package a camera and computing power comparable to the supercomputers of only a few decades ago (plus a whole slew of input sensors and output options). As a result they have made it technically feasible to enable gesture detection on readily available portable hardware, as demonstrated by software technology developed at Crunchfish.

However, the ability to detect gesture interactions is not enough on its own. Most of our current interfaces have been designed around other modes of interaction, which hints at a deeper problem: if some form of interaction is only feasible with gestures, then by definition none of the existing interfaces support this interaction. If there are unique qualities to gesture interactions, fresh designs and new insights are needed to do justice to them.

This thesis project aims to explore novel designs for gesture interfaces. Concretely it will describe a portfolio of designs produced for Crunchfish, to be used as exemplars. The theoretical work that will be produced is a discussion of gesture interfaces in general, using these designs as a starting point. This will include the topic of mediation and natural user interfaces, as reflected upon from a (post)phenomenological standpoint.

---

1 I must admit that I do not quite grasp what distinguishes post-phenomenology from phenomenology – my best guess is that it distinguishes phenomenological philosophy written before and after the advent of post-modernism. Regardless, since I cite P.P. Verbeek, who considers his work post-phenomenological, my reflection presumably is grounded in post-phenomenology as well.
2. The Research Focus

Crunchfish\(^2\) is a company in the city of Malmö, Sweden that has developed software capable of detecting gestures using the processing power and built-in cameras present in currently available smart phones and tablets. This gives Crunchfish an advantage in deploying this technology to the market, as their gesture detection does not require new special sensors. Furthermore, the technology is likely to only become easier to deploy and more reliable in the future, since the quality of phone cameras and available processing will continue to improve. At the time of writing their software, which is based on artificial neural networks, can be trained to detect hand shapes, transitions between these shapes, and the relative position and movement of the hand within the camera feed. Rudimentary depth detection exists by comparing changes in shape size, and it can detect the presence and location of a face, although it cannot distinguish whether a detected face is the same face as detected before.

Crunchfish does not focus on building applications with the technology it develops. Instead, it is licensed to third parties, including phone manufacturers. This means that it is in Crunchfish' benefit to have a catalogue of compelling examples showing use-cases of their technology. Therefore, the first goal of the internship associated with the thesis project is finding designs and applications that highlight the added value of using gestures compared to existing modes of interaction, using the technology developed at Crunchfish. These could be novel use-cases or existing ones redesigned to be enriched by the usage of gestures. The knowledge contribution of this work revolves around the design openings found with regard to these gesture interfaces, as well as a discussion on the notion of natural user interfaces in the context of gesture interfaces.

The technical constraints of the target hardware (tablets, smart phones and perhaps laptops) have helped focus the design process by a priori narrowing the design space. In theory this may also have limited the ability to find designs highlighting the strengths of gesture interfaces. However, tablets and smart phones are already liberating compared to previous gesture detection set-ups: they provide a novel, mobile frame of reference, where previous gesture detection systems have been bound to a

\(^2\) http://crunchfish.com/
The Research Focus

desktop or a room. A number of designs proposals play with this notion of the sensing device being small, portable and even wearable.

In order to deliver a portfolio within the eight week time-frame that is allotted to the project, it was important to be able to test and iterate ideas quickly. Because of that, the design proposals have only been tried out with mock-ups, using Wizard of Oz style settings and the like. This could bring the risk of them being technically infeasible with the hardware that Crunchfish’s technology runs on. In order to alleviate this, during entire design process a close dialogue with the technical staff of Crunchfish has been maintained to keep the designs technically grounded.

One trap that we wanted to break out of is looking at gestures in terms of other existing forms of input. It is unlikely that the unique qualities of a medium can be found by using as a proxy for another medium, except perhaps by reflecting on the points where it fails to adequately do so. For example, certain home theatre systems now have built in gesture detection, allowing the user to play, pause, manipulate the volume, etcetera. In this the gestures approximate the functionality of a remote controller. One advantage is that from the user’s perspective this removes the need for a peripheral input device (although technically we replace the remote controller with a camera sensor). One downside is that has to know the right gestures beforehand: a remote controller has icons to identify which buttons do what, and even if these symbols are unfamiliar to the user, the user can press a button to discover what it does.

Similar to the problems with using gestures to approximate other media, is that when the design of an interface focusses on the differences between gestures and other forms of input, it can lead to an interface implicitly designed around drawing attention to this difference, and hence to itself. Unless the interaction was designed around novelty, this is likely to be undesirable: one criterion of good interface design is that it stand as little in the way of the interaction as possible. It is important that gestures are approached as a medium of its own, with its unique strengths and weaknesses, and part of this thesis is therefore focused on discussing and breaking down what is meant when we talk about gestures.
3 Theory, related literature and previous work

This is my second thesis centred on gesture interfaces, and one might expect that I would build on the work of the previous one. However, my previous thesis focused on gesture interface design in relation to sign language, text interfaces and the use of language in interface design. As topics to design around, all of these were considered outside of the scope of what was feasible during the internship at Crunchfish. As a result, the previous thesis has not been used as a starting point for this project. However, given the strong overlap in related literature, I refer to (Van der Zwan, 2014) for a more detailed discussion of previous research on gesture interfaces and bimanual input. The rest of this section exclusively discusses the theory relevant to the work done during the current project.

3.1 Gesture Interface Research

The best starting point when doing research on gesture interfaces, including two-dimensional gestures, is probably the work of Maria Karam. To be specific, “A Taxonomy Of Gestures In Human Computer Interactions” (Karam, 2005) defines a clear vocabulary to classify gesture forms. Building on that work is her PhD thesis, “A framework for research and design of gesture-based human computer interactions” (Karam, 2006). It does a splendid job of giving an overview of the last forty years of research.

3.1.1 Classification of different gesture styles

Karam’s framework classifies five commonly used terms to describe different styles of gestures and gestures interfaces: deictic, manipulative and semaphoric gestures, gesticulation, and finally language. In the rest of this thesis we will use these terms according to these definitions, which are summarised below.

Deictic gestures “involve pointing to establish the identity or spatial location of an object” (Karam, 2006). Manipulative gestures are those gestures for which the movement of the hand tightly maps to the manipulated object – the movement of a mouse to the movement of the mouse pointer, for example.

Semaphoric gestures are symbols – a wave to say hello, a thumbs-up to approve. The examples highlight two different types of semaphoric gestures: dynamic (motion
based) and static (hand-shape based). Many gestures combine the two: waving uses an open palm with a specific motion. Just as there are no true universal words, there are no universal semaphoric gestures, although some can safely be presumed to be familiar to everyone. However, in the general sense a user is required to learn a vocabulary of semaphoric gestures before they can be used. Supposedly, semaphoric gestures are most commonly applied gesture style within gesture interfaces, despite being the least used form of gesturing by humans (Karam, 2006). On a personal note, I suspect that this is due to the aforementioned tendency to use gestures to approximate other input forms: a naive way to design a gesture interaction is to have it pantomime an existing interaction (say, a button push), which will result in a semaphoric gesture.

Gesticulation is “considered one of the most natural forms of gesturing” (Ibid.). Originally called ‘coverbal gestures’, they are “intended to add clarity to speech recognition, where a verbal description of a physical shape or form is depicted through the gestures” (Ibid.).

Language gestures could be said to be the result of combining deictic gestures, gesticulation and a large vocabulary of semaphoric gestures with a grammar, syntax to create a sign language.

As described in the research focus, Crunchfish’s technology can identify hand shapes, transitions between these shapes, and the relative position and movement of detected shapes within the camera feed. Therefore, it is best suited for manipulative and semaphoric gestures.

### 3.2 Bimanual Input

The ability to distinguish two hands is an advantage that Crunchfish’ technology has over a touch screen, which can distinguish touch points but not their sources. Therefore the question of how and when to use bimanual input is relevant to this thesis. Most existing papers on bimanual input are not directly related to gesture interfaces. However, many of the design principles they propose apply to bimanual interfaces in general. One particularly influential paper is “Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model”, by Yves Guiard (Guiard, 1987). The paper describes the kinematic chain model, a general model to break down bimanual actions and to reflect on how they are used in given contexts.
3. Theory, related literature and previous work

Fig. 1: Top: parallel action. Middle: orthogonal action. Bottom: cooperative bimanual action
3.2.1 The Kinematic Chain model

Guiard proposes to model the limbs as an abstract chain of motors, starting at the most coarse-grained level and progressing towards more refined motion with each link. This linked motion allows one to move with in large area due to the coarsest motor, but with great precision due to the most precise motors at the end of the chain (that is, if every link performs “error correction” for the lack of precision of the preceding links). For example, one can apply this model to an arm: the shoulder is the first motor, the most coarse-grained one. The elbow is the second motor, the wrist the third, the fingers the last. The kinematic chain allows us to have the sub-millimetre precision of our fingertips over the entire reach of our arm-span, which covers a space three orders of magnitude larger.

Furthermore, Guiard states that when our two limbs work together they can also function as two motors in a kinematic chain. In this situation, the hand that is the first motor is commonly considered the off-hand. However, it is actually the hand that sets up the frame of reference with crude, coarse motions. What we normally consider the dominant hand is the second motor in this kinematic chain, and it performs the refined, final motion. This is an understandable mistake, given that the refined hand is stronger, more precise, and the endpoint in the kinematic chain.

Fig. 2: Writing as a cooperative bimanual action.

On the left the result, on the right the actual motion of the pen.

Source: (Guiard, 1987)
Guiard identifies three different types of bimanual actions: parallel, orthogonal and cooperative (fig. 1). Parallel bimanual actions are those where both hands do the same thing in some symmetrical fashion – swimming is one example. Orthogonal bimanual actions are those where the limbs do separate tasks – holding something in one hand while opening the door with another. Cooperative bimanual actions are the aforementioned kinematic chain. For example, when handwriting the frame-of-reference hand moves the paper as the “dominant” hand writes (fig. 2). Research shows people write slower when the frame-of-reference hand is not allowed to hold and move the paper (Guiard, 1987).

Analysis of experimental bimanual designs has shown that when designs align with the kinematic chain model they are the most effective (Leganchuk et al., 1998). Moreover, complex actions where a frame of reference needs to be set up and adjusted while the main task is performed, are likely to be more effectively performed bimanually than unimanually. Furthermore, it is important the right role is assigned to the right hand, even if the task is not bimanual: assigning frame-of-reference-related tasks to our dominant hands or vice-versa will likely be considered unpleasant.

3.3 A link between design and (post)phenomenology

Phenomenology is a branch of philosophy that “[studies] structures of consciousness as experienced from the first-person point of view” (Smith, 2013). It is too broad a philosophy to discuss here in full, and I profess that my understanding of it is limited. However, I believe that its notions of mediation and embodiment can be valuable to interface design and the discussion of so-called “natural” user interfaces, a topic that is often considered a logical match for gesture interfaces. Furthermore, approaching design from a phenomenological starting point can help to minimise making implicit assumptions about what, for example, a smart phone is. This is valuable when exploring a new design space, such as gesture interfaces.

3.3.1 Mediation & Embodiment

The Stanford Encyclopedia of Philosophy suggests that from an etymological point of view: “phenomenology is the study of phenomena: literally, appearances as opposed to reality” (Smith, 2013). It studies realities from the starting point of experience. Because the possibility to have an experience is rooted in the body, a great deal of emphasis is places on the senses, mediation and embodiment.
In phenomenological terms, we are not directly connected with the world, but can only access “our” world, which is “the world as it is disclosed by us”. It is mediated by a medium through our senses. For example, one person can mediate thoughts about a light bulb through sound using verbal communication. These sounds then get sensed by the ears of a recipient, and (potentially mis-)interpreted in the brain (fig. 3). Note that this written text describing this is as well as the above illustration are themselves forms of mediation.

As another example, let us describe what happens when we observe an image of a flower on a screen. Technically, we are not looking at a flower. We are looking at a pane of glass, behind which lie a number light-emitting cells that can change colours called picture elements or pixels. Our visual senses are focused on these pixels, but often we do not see them as much as look “through” them as a whole to see the represented image, in this case a flower. The screen is an additional layer of mediation between us and the direct presentation of the flower.

Embodiment can be discussed in multiple ways. One is the mental extension of the physical self and the physical senses to objects beyond our body (fig. 4). We embody
items we wear or handle. Note that we can combine sensory embodiment with Guiard’s model to think in terms of extending our kinematic chain beyond our limbs. One example of this is found Feynman’s famous “There’s Plenty Of Room At The Bottom” lecture, in which he suggests an amusing way to work our way down to the nanometre scale:

I want [robotic hands] to be made [...] so that they are one-fourth the scale of the “hands” that you ordinarily maneuver. So you have a scheme by which you can do things at one-quarter scale [...] Aha! So [using these robotic hands] I manufacture a quarter-size lathe; I manufacture quarter-size tools; and I make, at the one-quarter scale, still another set of hands again relatively one-quarter size! This is one-sixteenth size, from my point of view. And after I finish doing this [...] I can now manipulate the one-sixteenth size hands. Well, you get the principle from there on. (Feynman, 1960)

![Fig. 4: Mediation and Embodiment](Illustration by Jaffar Salih)
Feynman essentially suggested that we extend our limbs all the way to the nanometre scale by embodying this chain of “robotic hands” as a continuation of our “natural” kinematic chain.

Yet another way to look at embodiment is in terms of thinking with the body, as opposed to thinking purely with the brain:

*Many features of cognition are embodied in that they are deeply dependent upon characteristics of the physical body of an agent, such that the agent’s beyond-the-brain body plays a significant causal role, or a physically constitutive role, in that agent’s cognitive processing.* (Wilson and Foglia, 2011)

One can combine these two views and reflect on how artificial extensions of the body affect our embodied cognition. For example, how we change our identity and behaviour to match the clothes we wear and literally embody social roles (fig. 4).

The philosophical notions of mediation and embodiment model phenomena with real consequences to how we connect to what we see, and it is worthwhile to reflect on this during our design process. For example, in the case of a smart phone or tablet, the way we commonly interact with the representations on our screen is by pressing the tip of our finger onto the screen. Bret Victor wrote of this as follows:

*I call this technology Pictures Under Glass. Pictures Under Glass sacrifice all the tactile richness of working with our hands, offering instead a hokey visual façade. [...] Pictures Under Glass is an interaction paradigm of permanent numbness.* (Bret Victor, 2011)

Victor was discussing how touchscreens undermine capabilities to use the tactile senses of our hands – how this form of *mediation* thwarts our *embodiment*. This is especially relevant for gesture input, as it has no tactile feedback, being completely touchless. An important question to ask then is what effect this has on the interaction in question and on the overall experience, and whether or not this is a problem given any particular use-case.

### 3.3.2 Technical Intentionality

Peter-Paul Verbeek and Petran Kockelkoren have looked at design from the standpoint of (post)phenomenology. The papers “The Things that Matter,” (Verbeek and...

One of their suggestions is that artefacts have within them a form of “technological intentionality” – what they enable (or prevent) guides or “co-shapes” the behaviour of the user (Verbeek and Kockelkoren, 1998). Take for example the design of the original command line interface: the best interactive feedback loop a computer had to offer in those days was keyboard input and line printer output. Its interface design thus was shaped around this feedback loop. Technological intentionality suggests that the behaviour of the user on such a given system is then shaped by the system as a whole, and that this determines the further development of the system.

To give another example: in graphical user interfaces for the desktop, most interface elements are interacted with through a mouse, trackball or trackpad. What these three forms of input have in common is that the on-screen pointer first has to be moved in the general direction of the item that is to be interacted with. This is a fairly coarse action – a relatively large distance may have to be covered on the screen. Only when the item is reached do the movements become precise\textsuperscript{3}. As a consequence, one type of interaction that works well is lining up interactive elements along the edge. This allows the users to “slam” the mouse pointer towards the edge of the screen, then slide along the edge of the screen to select an item. Many interface items in desktop software are arranged to work well with this: the taskbar that most operating systems use, the menu items in many programs being situated at the screen edges, etcetera. In fact, many “auto-hide” functionalities rely on the user “pushing” the mouse pointer into the edge of the screen. However, with touchscreens we do not move a pointer, we move our finger or stylus, which is not bound by the edge of the screen, nor is capable of pushing into the edge. This means that the behaviour described above does not work, and other types of menus, like pie menus or menus that can slide in from the outside, are often more appropriate.

Technical intentionality is somewhat posed as a counter to Platonically framing artefacts as material derivatives of an idea, and complementary to Madeleine Akrich’s\textsuperscript{3} Note how this overlaps with the physical motion of the arm involved, which is a kinematic chain that starts with coarse arm movements and ends with fine grained nudges by the wrist.
3. Theory, related literature and previous work

The notion of “scripts” that are “inscribed” into the design of objects. For this thesis the relevant question is: what are the technical intentionalities of gestures? How does the medium of gestures shape the behaviour of the user? Presumably, answers to this question will guide us to more appropriate uses of gestures in design, and in fact has inspired the starting points for certain explorations during this project.

A problem with technical intentionalities is that they must co-shape the behaviour of the user before they can be observed. As gesture interfaces are not widely used yet, this behaviour has yet to emerge. However, there is one community who has a lot experience with gestures: the Deaf, through their sign languages. By looking for the differences between Deaf and mainstream culture that are not a consequence of being deaf, but of the differences between audible and sign languages, we may identify technical intentionalities of using sign languages to communicate. The assumption is made that this technical intentionality would then hold for gesture interfaces too.

One thing mentioned in the literature is that signing is not affected by nearby signing in the same way that audible languages are affected by nearby sounds (Sacks, 1990). With audible languages sounds overlap other sounds. Speaking across a full room requires shouting over the background murmur, likely interrupting the conversations of others. In contrast, as a consequence of their visual nature, gestures are only affected by things obstructing their view. Provided clear visibility, the Deaf can sign across large distances without interrupting anyone else. Similarly, eavesdropping a conversation is as simple as looking at a person. As one would expect, this difference in the behaviour of the communication medium compared to audible speech has subtly shaped cultural norms for the Deaf.

From this we can conclude that a technical intentionality of using the medium of gestures is that they have a visible, readable and even performative nature. However, unlike sound they do not risk overlapping with other visual input when one is not directly looking at them. Applying this to the context of a phone, think of holding a conversation through the phone in a public place. When doing so by typing out text messages, the content of the conversation would be invisible to the surroundings. On the other hand, audibly talking on the phone it is not only easy to follow, but even hard to

---

4 There is a convention in the literature to distinguish Deaf who can sign from those who cannot by capitalization.
ignore in some cases. Gesture input sits somewhere in between these two, being publicly readable but not very intrusive.

Interfaces using gesture input would do well to ensure their use-case aligns or at least does not conflict with this technical intentionality. When privacy is required, gestures are likely inappropriate, for example. Interestingly, we can already see this in play out in the way (fake) gesture interfaces are depicted in science fiction films. An easy criticism of these is that they are not designed or used so much with usability in mind, but as a form of visual exposition. In fact, usability is often sacrificed in the service of dramatic performance. However, it is the visually readable, performative nature of gestures that is the main reason why they are so popular as interfaces in science fiction films in the first place.

3.3.3 What postphenomenology implies regarding “natural” user interfaces

Historically, gesture interfaces have been presumed to be a good fit for so-called “intuitive” or “natural” user interfaces. This has often been in the form of multimodal interfaces where gestures and voice-activation are combined, such as the famous “Put That There” installation by Chris Schmandt (fig. 5). An underlying assumption is that trying to make an interface more “intuitive” and “natural” requires making it more like human-to-human communication. For most people human-to-human interaction is the first and most familiar form of communicating intent, and it can be considered “intuitive” in that specific sense. Similarly, designing for a human body plan brings certain expectations in terms of physical ergonomics, and the aforementioned finding that bimanual designs aligned with Guiard’s kinematic chain model are more effective can be interpreted as cognitive ergonomics. However, what it means to be more “natural”, let alone whether or not it is better, is debatable. It is often treated as if it is a determinable static ideal, followed by the (implicit) assumption that designing a natural user interface is a matter of getting all the “unnatural” parts out of the way to get closer to this “natural foundation.”

Petran Kockelkoren has argued that from a phenomenological perspective there is no such thing as a “natural foundation”, because “[humans] do not coincide with themselves. [...] They are able to distance themselves [and] even stand beside themselves and look over their own shoulder, as it were, at everything they do. People are
outsiders in relation to themselves” (Kockelkoren, 2005). He called this phenomenon “human ex-centricity”, a term that was first described by the phenomenologist Helmuth Plessner in the 1930s, but mostly forgotten outside of philosophy circles. Kockelkoren concludes that the paradox of fundamentally being *distanced* from the self implies that there cannot be a “natural foundation” to get closer to. Instead, he suggests that we are “naturally artificial” and that we are always in a dynamic state of reinventing our naturally artificial foundation: “[There] is no underlying, original substratum. There is only a permanent oscillation between decentering and recentring, with mediatory technologies as the engines of change” (Ibid.) This implies that
the determinable static ideal is a flawed premise, as is designing by getting all the “unnatural” parts out of the way. Kockelkoren states: “recentring does not lead us back to some unspoiled, primeval state, but at most it brings about a temporary state of equilibrium in a process of technological mediation.” Technology shapes us and is shaped by us, and in the process we reinvent what we consider natural.

The notion of human ex-centricity was reintroduced in a paper discussing the shared role of scientists and artists (and by extension designers) in “domesticating” new technologies. They all share the role of pioneers exploring the frontier of new technologies, that at first tend to be considered unnatural for mainstream use. The phenomenological mindset of “naturally artificial human ex-centricity” frames interface technology not as something that should try to approximate a timeless natural human foundation, but as a process of taming the wild and unexplored. In this light, gesture faces are not “natural” or “intuitive,” but an unfamiliar, unexplored technology yet to be domesticated by both designers and users, yet to be “made” natural.

3.4 Theory – Summary
The concepts of semaphoric, deictic, manipulative gestures and gesticulation, combined with Guiard’s kinematic chain model, provide a clear vocabulary to break down and analyse gesture interfaces, bimanual or otherwise. Further engagements with theory during this project mainly consist of looking through a phenomenological, “human ex-centric design” lens, both when reflecting on the explored designs and when looking for new inspiration. Gestures have long been assumed to show great promise for “natural” user interfaces, a notion that has indirectly been challenged by Kockelkoren’s discussion of human ex-centricity. Furthermore, interaction and mediation are closely related, therefore insights from this post-phenomenological approach to artefacts and mediation are likely to help in finding novel ways of looking at gestures, leading to novel designs. Letting go of the notions of “natural” or “intuitive” interfaces has let me approach gestures with a fresh mindset. While this chapter may leave the impression if I have approached the topic from a high level, the design process itself was a very hands-on engagement with the material. I have been applying and trying out these concepts alongside insights from earlier gesture interface research, general interface research and fields related to gestures, such as sign language.
4 Methodology

The project is fundamentally exploratory in nature. Because of that my thesis supervisor (Jonas Löwgren) and I quickly came to the realisation that the most appropriate methodology would be one inspired by programmatic research practices. In one of my last discussions with Löwgren about how programmatic research has inspired the methodology of the project, he paraphrased the process in such succinctly, accessible terms that I will cite it in full:

*My project is not about defining a problem, then solving it. It is not even about formulating a hypothesis or research question, then answering it. It is rather a design-led exploration of a field of possibilities. In this exploration I experiment with ideas in the forms of sketches and prototypes. I also think quite a bit, combining thoughts on my experiments with other people’s thoughts that I find in academic literature. This thinking helps me decide what the next experiment should be. My “result” is the sum of what I have found out about the field of possibilities, including experiments and thoughts. Since this an academic setting, I also report the grounds for my findings such that other academics can judge their credibility. The result is a snapshot of what I have learnt at the time of writing. If I were to work for another month on the project, there would be a different snapshot that is not necessarily incremental compared to the present one – it could happen that I learn something that makes me reframe the exploration. This approach is inspired by what I understand “programmatic research” to be.*

My understanding of programmatic research practices is mainly based on how it is described in “Towards Programmatic Design Research” (Löwgren et al., 2013) and “Designing for Homo Explorrens” (Hobye, 2014). Programmatic research is a design research methodology. It treats the design practice itself as an integral part of knowledge production. This is illustrated in fig. 6, which describes the programmatic research process and can be interpreted as follows:
4. Methodology

- **Optics**: The theoretical framing of what we do, will be doing and have done. This partially determines what we will observe, because it determines where and how we “look.” Of course, what we will observe also depends on what is looked at, but our existing theoretical framing shapes our perception.

- **Engagements**: These can be engagements with the materials, with the design space, the world of concrete ideas and possibilities – they do not have to be “empirical” in the conventional sense of building a functioning prototype and user-testing it. In short, what we “do”. In turn, the results of what we “do” changes the way we “look” (the optics).

- **Drifting**: As mentioned above, programmatic research does not start with a clearly defined, fixed research question, or a clearly delineated problem to be solved. It has an open-ended exploratory nature. Because of that it is important to let the design process go wherever the findings seem to lead it, to let ideas and insights from one design lead to another. This is defined as drifting.

- **Takeaways**: As we drift across the design space, we report on the lessons learned from this constant back and forth process between optics and engagements. These are the snapshots of the process mentioned by Löwgren, and by

---

*Fig. 6: “The optics form the programmatic frame, the engagements challenge this frame and make it drift, and the takeaways are static snapshots of the current state of the overall program.” – M Hobye

*Source: (Hobye, 2014)*
looking back at a number of them for a longer period of time, we might observe an overall grander theme that we drifted across. The latter type of takeaway may indicate a more general insight than the typical snapshots these processes produce.

It should be emphasized that my thesis is inspired by these practices, but not remotely large enough in scope to reasonably be considered a complete program. True programmatic research takes place over a longer period: it takes a large amount of engagements and snapshots for the drifting to “stabilize”, as it were.

Interestingly, one can see similarities between the approach taken in programmatic research and Kockelkoren’s notions of human ex-centricity and humans being naturally artificial. “A permanent oscillation between decentering and recentring, with mediatory technologies as the engines of change” sounds suspiciously similar to drifting across a design space, caused by the constant back-and-forth between optics and engagements. Similarly, the stabilization one might achieve after drifting through engagements with one particular topic for a long time, could be described as Kockelkoren’s temporary state of equilibrium as the result of recentring. Perhaps new insights could be gained from exploring if and how these two discussions connect more deeply. For this project however, it suffices that it has been fruitful in the design process to add Verbeek and Kockelkoren’s writings to the theories used in my ‘optics’.
5 Portfolio – the exploratory Process

Recall that the entire project took place over the time-span of eight weeks. The approach we (meaning Crunchfish and I) took to exploring the design space was to look broad and deep into various potential design openings – using our ‘optics’ to choose where to engage – and continuously try to discover new design openings as things were tried out. Our hope was that halfway, near the end of week three or beginning of week four, a number of designs worth developing further would have been identified this way. These would be the designs that were both novel and interesting, and aligned well with Crunchfish resources. Starting in week five and onwards we would then focus on polishing and refining these designs that showed most promise.

This approach appears to have worked out in practice, as we mostly stuck to this initial plan. However, I must caution that due to my previous experience with gesture interfaces I could immediately jump into the subject. Someone new to gesture interfaces would probably require more time to investigate the theoretical background of gestures as a medium. However, because of my previous work on gestures I was able to skip that slow initial step and be off to a running start.

5.1 Week 1 – Opening up the design space

One of the issues with designing gesture interfaces seems to be that it is really tempting to think of gestures as a technology that can simply replace an existing form of interaction: “click on the appropriate button to change song,” with another: “swipe right or left to change to the next of previous song.” This retrofitting of gestures onto existing interfaces has a number of issues associated with it. First of all, the existing interfaces are often optimised specifically for the current forms of interaction. Gestures are often a poor substitute for the latter in such a context and do not clearly offer added value, unless we count “novelty”. Second, this is not a good way of discovering what makes gestures unique, as being treated as a proxy for other forms of interaction focusses on how they are like those forms of interaction. Third, taking a function and choosing a way to trigger it is the end of a long chain of design decisions, starting with what it is that we really want to do: “enjoy listening to music,” to how we achieve this: “allow users to listen anywhere,” which leads to: “use a mobile platform,” which leads to: “using a phone or tablet.” Similarly “let users choose their preferred
music” leads to: “provide possibility to change songs,” which leads to the aforementioned: “switch to next/previous song.” These design choices are full of assumptions (often implicit) about the context within which the application is used, and it is this very context that is changed by the presence of a new form of input. All of these design choices then have to be re-evaluated, lest we end up with more-of-the-same.

5.1.1 THE SMART PHONE AS A CASE-STUDY OF IMPLICIT DESIGN ASSUMPTIONS

Let us look at some of the design contexts within the domain of smart phones, where Crunchfish’ technology is commonly applied. Being mainly focused on existing mobile applications leads to many, often unnoticed, assumptions about how a phone is used. The phone can be manipulated with one hand or bimanually. When the phone is used with one hand, this is likely to be the dominant hand, with the screen being manipulated by the thumb. When used bimanually, one hand usually holds the phone while the index finger of the other hand manipulates the touch-screen – which role is assigned to the dominant and frame of reference hand depending on the circumstances (for example: is the input bimanual from the start, or is a switch made from
unimanual input?) There are some patterns among the exceptions to these common ways of holding and interacting with the phone, for example when the interface mimics a game controller. In this case, the phone is often held sideways, with either hand holding a side, and on both sides of the screen interface elements will be found that can be manipulated by the thumbs (fig. 8). However, most mobile apps effectively replace a single mouse cursor with a single finger pressing a button, and do not fully use the possibilities of a multi-touch screen. Most examples previously developed by Crunchfish essentially replace this single-finger button input with a gesture trigger, leaving the design space relatively unexplored.

The consequence of starting at the end of all of these conventions is that we go along with countless implicit design choices that were made along the way, without exploring alternate possibilities. These design choices have numerous subtle consequences when adding interaction based on gesture detection. Below some of these choices are listed, together with some alternative options.

- **Most mobile applications assume single-handed input.** The touch screen is not able to distinguish what is touching it – it could be the tip of a nose for all we know. This limits the ability to design bimanual interfaces, requiring specially designed layouts like the emulated gamepad in fig. 8. In those situations we can make assumptions about how the phone is held and which digit is responsible for input. The ability to distinguish gestures from touch input gives us more options. For example, we could design around the assumption that one hand gestures while the other touches the screen.

- **The hand that holds the phone determines what hand performs the gestures.** When retrofitting, it is likely that the touch screen interface is implicitly designed around manipulation by the dominant hand. If the dominant hand holds the phone, possibly manipulating the touch-screen by thumb, the gestures must be performed by the frame-of-reference hand. If the frame-of-reference hand holds the phone while the dominant hand performs the touch input, the dominant hand must also perform the gestures. This difference influences what the type of gesture input feels “appropriate” for a given task, and which type of bimanual interface design works best. Note that this only applies if the phone needs to be held in the first place.
• **Most interaction is designed around explicit screen-based interaction and feedback.** Existing applications are often designed around touch screen input. Given this interaction with the screen, it common to use it as the main source of feedback as well. Yet, similar to how a touch typist does not need to look at the keyboard, using the touch screen as our main form of input does not (technically) require our visual attention. However, as pointed out in (Bret Victor, 2011), beyond making contact with the screen, a touch screen provides essentially no tactile feedback. Gesture detection has no inherent feedback whatsoever, depending even more on external sources. However, since gesture detection works independently from the screen, we also have more freedom to choose any of the available forms of feedback.

• **The smart phone or tablet is a portable mini-computer.** Another example of approaching a medium as a proxy of another, smart phones tend to be used for similar tasks to that of the desktop computer (consuming media, reading email). However, the modern smart phone is capable of detecting sounds, sights, touch, motion, orientation and (using GPS) location relative to the planet, and aside from audiovisual feedback has limited haptic feedback possibilities in the form of vibration. If we frame the smart phone or tablet as a small, portable sensing device with an attached computer to process the data, we open up an enormous design space.

Not being aware of these design choices results in implicitly going along with the conventional answers. If we start “earlier” in the design process, before these choices are made, and consciously decide to go one way or the other, our the design space is much larger. This is important if we want to escape the local design optimum, which is not geared towards gesture interfaces.

5.1.2 **User scenario: social exploration in the context of grandparents and their grandchildren**

All of the discussed insights above are still high-level and abstract, so to actually apply them we started with a simple user scenario. The idea was that a concrete setting would help inspire some first designs. The scenario was that of social exploration in the context of grandparents with their grandchildren. It was chosen because this type of social interaction appears particularly vulnerable to technology. One reason for
this is different levels of comfort with using and exploring new technology. Another issue is that computer interaction is not always particularly readable to third parties: it can be hard for grandparents to follow what their grandchildren are doing on their mobile device, and vice-versa. The performative aspect of gestures could mitigate the latter issue. Furthermore, Crunchfish’ gesture detection can, to some degree, detect multiple gestures at once. To be precise: when it is set up to detect semaphoric gestures, it will track any matching hand-shapes it is currently detecting. With the right interface design this would allow for input from multiple users, which opens up the design space for social and collaborative interactions. Taking into account the normal social context and “functions” of grandparent/grandchild interaction, the aim then is to come up with designs such that:

1. Social exploration and social bonding is nurtured.

2. Physical exploration of the environment is encouraged, as exploring the house, neighbourhood and larger world is an important element of grandparent/grandchild interaction. The designs should therefore draw the attention to the physical here and now, instead of allowing the users to “disappear” into the mobile device. Note that we might exploit the portability of the technology in question here.

3. It allows for the passing on of traditions and culture. While more difficult, this is an important value of grandparent/grandchild interactions and should be on our minds during the design process.

5.1.3 DESIGN CONCEPT: THE TABLET AS A MOBILE KNOWLEDGE DATABASE

It has long been known in developmental psychology that the way in which a parental figure communicates with a young child has an enormous impact on the child’s eagerness to explore and take initiative later in life:

[An active and questing disposition in the mind] is not something that arises spontaneously, de novo, or directly from the impact of experience; it stems, it is stimulated, by communicative exchange – it requires dialogue, in particular the complex dialogue of mother and child. [...] A terrible power, it would seem, lies with the mother to communicate with her child properly or not; to introduce probing questions such as “How?” “Why?” and “What if?” or replace them with a
mindless monologue of “What's this?” “Do that”; to communicate a sense of logic and causality, or to leave everything at the dumb level of unaccountability; to introduce a sense of place and time, or to refer only to the here and now; to introduce [...] a conceptual world that will give coherence and meaning to life, and challenge the mind and emotions of the child, or to leave everything at the level of the ungeneralized, the unquestioned, at something almost below the animal level of the perceptual. (Sacks, 1990)

One can look at interfaces and their target audiences, and think about what social interactions and power dynamics they might encourage. With this in mind we decided to look at map applications, bird databases to complement birdwatching and plant database apps for gardening. Encyclopedic interfaces such as these typically require reading and writing skills to navigate, excluding smaller children when using them. Furthermore, the function of the application is only to be a reference point and guide for something that exists the actual world out there. This would meet the second requirement of the list above. To start with an activity complemented by these applications we imagined a scenario where grandparents and their grandchildren go bird watching or identifying plants together in the garden, or during a walk through the forest. Using the camera present in a mobile phone or tablet they could take snapshots of plants and movies of birds together. This would also automatically store information about the location and time the bird or plant was seen.

Imagine that this tablet has a birdwatching app or a plant database. One issue with traditional encyclopedia is that one needs to know the name of what is being looked up, which is the opposite of trying to figure out what bird or plant one just saw. One way to solve this is to ask the user to list specific features, which will narrow the possible options until one species is left. Let’s imagine the app can identify plants and birds like this by asking yes-or-no questions: “are the leaves shaped like this?”, which would be accompanied by pictures. The application would let the users answer “yes” or “no” with a thumbs up or down. In a small group the application allows people to vote together on an answer this way. After a number of questions, combined with location data from the phone, the app could identify with reasonable certainty what bird or plant they actually found, which would then be followed by interesting information about these how these birds live or what special properties the plants have, making full use of the rich media capabilities of the phone/tablet.
This imagined app obviously takes a retrofitting approach, adding gesture-based public voting to knowledge database applications that could already exist. However, there is reason to believe that even this small change would affect the social interaction when using this application. Compared to, for example, a grandparent being the only person capable of navigating the knowledge database and telling the child what bird they saw, this interface requires that they agree on what they saw, which would hopefully nudge its users into a dialogue. The result would be a more collaborative interaction. If the questions and answers are spoken out loud, the interface would even allow young children to explore this knowledge database on their own.

5.1.4 Design proposal: the smart phone as a wearable accessory to augment play

A second design opening was centred around exploring with the mediated nature of the screen, which resulted in a handful of game designs.

5.1.4.1 The Minotaur’s Tail

The first game design revolves around using the phone’s screen as an added layer of mediation between the player and the surrounding world, and turn this into a game mechanic. Imagine a (grand)parent playing a Minotaur, slowly lumbering about trying to catch the grandchildren. The other players have to steal the tail of the Minotaur.
Phone is attached to a blindfold as a makeshift ‘wearable’

Screen shows an animated eye, becoming a cyclops mask

If a face is detected by the camera, phone vibrates as haptic feedback for the player

Fig. 9: The Blinded Cyclops – the phone as an artificial extension of the senses

without being seen. The Minotaur wears a mask. Instead of the normal peepholes to see through it holds phone on the inside. Effectively, the Minotaur can only see what the phone "sees", turning the mediated nature of the screen into a game element. For children playing as the Minotaur, this could be a way of exploring their senses through play. “Seen” in this context means being caught by the camera, which triggers on face-detection. Because of this, a player can also hide (according to the rules of the game) by hiding his or her face. In other words: by playing peek-a-boo.

We performed some preliminary testing of the concept with a low-fidelity prototype. The prototype consisted a paper bag with a hole punched into it and the smartphone’s default camera app (fig. 8). The aim was to experiment with the indirect, mediated nature of navigating the surroundings, which is a core element of the game. Seeing the surroundings through the camera screen deprived participants of depth vision and limited the field of view significantly. The consensus was that the experience of sensory deprivation forced the participants to fall back on their other senses. The limited horizontal field of view caused a lot of dramatic head-turning. While there appeared to be potential in this game design, it was not further pursued.

5 The first draft of this thesis was written a month before the unveiling of Google Cardboard, which as an accessory is a natural fit for this game concept.
5.1.4.2 The Blinded Cyclops

From this starting point we came up with a game concept we called called “The Blinded Cyclops” (fig. 9). In this game, the phone is attached to a blindfold (we used touch fastener straps). Thematically, the blindfolded player is now Polyphemus the Cyclops, while the other players are Odysseus and his crew trying to escape. This is played out as a game of tag with a time-limit. Whenever another player is detected, again using facial recognition, the phone vibrates. The Cyclops has to rely on other senses than his visual one. The face detection, sensed through haptic feedback of the phone, replaces his own sense of vision. For the other players, the screen on the phone can be used to give feedback of whether or not the Cyclops “sees” you, by having a large looming animated eye that changes expressions (fig. 10).

Note that every smart phone can function as an eye. We therefore are not limited to playing a cyclops; there are more monsters in Greek mythology that work. We could create a form of “surround vision” by attaching these surrogate eyes at various places on the body, as long as the haptic feedback can be sensed by the player. This
lets us play as the Argos Panoptes, the hundred eyed all-seeing giant, by using multiple smart phones in such a manner (although attaching one hundred smart phones would be a bit much). Another option would be to hold the phone in our hand as a portable eye and pretend we are one of the Grey Sisters. A simple program, consisting of nothing but an animated eye and haptic feedback being given when a face is detected, is thus turned into an asset for mythological role-playing, aiding the transmission of cultural knowledge in the process.

5.1.4.3 A futuristic heads-up display

A third design takes yet another spin on forcing the visual sensing of world outside of the mask to be mediated through the screen. We take a simple “robot suit”, for example made from a cardboard box (fig. 11), and add the phone as an intermediate screen. On this screen we use a handful of filters, combined with hand-detection and face-detection to create a so-called heads-up display as is commonly seen in science fiction movies (fig. 12). This would essentially be a form of augmented reality. However, unlike many AR designs, this only serves to increase the immersion of the child in his or her own made-up narrative, who for example might pretend him- or herself as a robot on a space mission, exploring an alien planet.

What all three of these designs have in common is an exploration of the body through both sensory deprivation and an artificial extension of the senses through the sensors of the phone and its capabilities to give feedback. It also turns the phone into a wearable. These are examples of how the design space opens up if we don’t start with the idea...
of a phone and all its implicit design choices and assumed use contexts. They do so by treating the phone as a portable sensing device, an intermediate point in the larger context of mediating the world.

5.2 **Week 2 – Investigating Collaborative Interfaces**

Because of the social readability of gestures, I investigated in what contexts this might be a significant benefit. One direction investigated was that of computer-supported cooperative work (CSCW). Various papers on CSCW were investigated (Bly, 1988), (Heath and Luff, 1992), (Dourish and Bellotti, 1992), (Kuzuoka et al., 1994), (Ou et al., 2003). Many of these appear to focus on the computer aiding in remote collaboration – a theme that can be traced back all the way to Douglas Engelbart’s Mother Of All Demos. Another theme among these papers was the need for users to be able to read the intent of their collaborators, both while performing an action and before it.
Another need was to be able to read what currently held the collaborators attention. With interfaces that do not meet these needs, users are limited to responding to each others actions in sequence. Interfaces that do meet these needs let the users anticipate each other's actions, allowing for simultaneous cooperative actions. One could even draw a parallel to Guiard's cooperative bimanual actions here! The readability of gestures suggests it might be an enabling mode of interaction in these situations.

5.2.1 Crisis Management Environments

One direction investigated was that of collaborative work environments such as control towers at airports and the line control rooms of the London Underground, based on the work by Christian Heath and Paul Luff. As described by Heath & Luff, these are multimedia environments in which personnel are effectively in a constant state of crisis management, with various responsibilities where they have to anticipate, respond to and resolve problems as they happen. There is little to no time to plan ahead, because the work fundamentally consists of responding to things not going as planned. Because everything has to be decided in the moment, members of personnel are highly autonomous, yet have to somehow coordinate every action they take with their co-workers the moment they take it. This need puts high demands on their ability to communicate intent:

The ability to coordinate activities, and the process of interpretation and perception it entails, inevitably relies upon a social organisation; a body of skills and practices which allows different personnel to recognise what each other is doing and thereby produce appropriate conduct. [...] we might conceive of this organisation as a form of 'distributed cognition'; a process in which various individuals develop an interrelated orientation towards a collection of tasks and activities (cf. Hutchins 1989, Olson 1990, Olson and Olson 1991). And yet, even this [...] does not quite capture [the] character of cooperative work. It is not simply that tasks and activities occur within a particular cultural framework and social context, but rather that collaboration necessitates a publicly available set of practices and reasoning, which are developed and warranted within a particular setting, and which systematically inform the work and interaction of various personnel. (Heath and Luff, 1992)
This type of work requires a lot of bottom-up coordination, a highly effortful act that involves a lot of explicit and implicit communication of all parties involved. The inherent readability of gestural input could be a boon in these circumstances, as the act of doing work would double as signalling to your co-workers what you are doing. However, these are also highly specialised niche situations – to come to a sensible design would require a concrete use-case and heavily involve participatory design practices with the user-group, and close observations of existing work environments. Given the time-scale of the project pursuing this option further was therefore considered out of the scope of this thesis.

5.2.2 A Collaborative Remote Whiteboard

Another direction investigated was to see if gestures could augment the Clearboard design prototype (Ishii and Kobayashi, 1992). At its heart Clearboard is a remote collaborative draw/sketching tool for two people. It is a deceptively simple concept over two decades old that nevertheless has yet to be implemented in commonly accessible remote collaboration tools. The design uses the metaphor of drawing on a glass window, with the collaborator sitting on the other side. For the digital version this would be achieved with a two-way video system (fig. 13). The participants see each other’s mirror image, because otherwise the two participants the writing of one would be mirrored for the other. What this the video feed adds over simply sharing a digital drawing surface remotely, for which a number of tools exist, is the constant awareness of what currently has their partner’s attention. Ishii and Kobayashi dubbed this feature “gaze awareness.” As mentioned before, the ability to able to read the partner’s intent makes an enormous difference in the smoothness of any collaboration, regardless of whether it is remote or not. In theory the Clearboard design is even more efficient than collaborating on a regular local whiteboard, because it removes the need to switch attention between the collaborator and the board.

Fig. 13: “Towards Seamless Collaboration Media: From TeamWorkstation to Clearboard”

Source: (Ishii and Kobayashi, 2012)
Gaze awareness goes a long way to allow partners to anticipate each other’s actions, and thus implicitly synchronise interactions with the application. In the case of Clearboard these are limited to drawing and erasing. However, this form of shared-screen collaboration might be generalised to other applications, such as the collaborative work-environments described by Heath & Luff. In the case of these more complicated use-contexts, intent might not be so easily readable from gaze awareness alone. Again the performative nature of gestures might help out.

While the potential benefits here were clear, it was decided to not explore this design space further within the project, for two reasons. First, like with the control rooms described by Heath & Luff, the jump from a conceptually useful design direction to a concrete design is difficult to achieve without a defined use-case, and it was hard to come up with one beyond the original Clearboard concept. This lead to the second reason, which is that the Clearboard concept cannot be implemented on readily available hardware. This is due to a straightforward but hard to solve technical matter: to create visual feedback that allows for gaze awareness, the set-up requires that the fields of view of the recording cameras have go through the screen (fig. 13, middle picture). Ishii’s team used a half-mirror to achieve this, but until transparent see-through screens are commonplace this will not be easily duplicated by off-the-shelf hardware. Since Crunchfish focuses on currently readily available smart phones and tablets, this was not a feasible direction for this project. Despite not developing this specific concept further, it did lead to a discussion of the placement of the camera on a smart phone or tablet, which lead to another design direction that was pursued in the following weeks.

5.3 Week 3 to 5 – Virtually Extending the Screen
The investigation of Ishii’s ClearBoard lead to the insight that a camera has a certain field of view. At the distance gestures are best detected, this field of view usually encompasses a (virtual) surface area greater than the size of the screen. This is something that we can play with. Two examples of potential applications were identified: a grab & drop clipboard that works by dragging/dropping items to and from slots “outside” the screen, and an advanced drawing application that hides all of its complicated options off-screen, until a gesture pulls them in. These were worked out into a more detailed storyboard and mock-up respectively over a number of weeks.
5.3.1 Design proposal: Grab & Drop clipboard

The first design space explored was that of using gestures in a scenario where objects could be grabbed and left out of the edge of the screen for later retrieval when they were needed, similar to leaving real-life objects around the desk within hands reach. We quickly realised that this was functionally the same as the copying/cutting and pasting paradigm, which also relies on stashing (a copy) of a virtual “object” away for later use. However, our design would treat these objects more like real-life object, allowing for more items to be stashed away and granting the ability to rely on spatial memory to retrieve them. Figures 14 to 17 depict an interface mock-up, showing a storyboard of how this program could be used in the context of a personal computer or laptop using a standard desktop interface.

Figure 14: We start with an empty rich text editing environment, in this example the LibreOffice Writer program (fig. 14a). The user lifts left hand from the keyboard, makes its palm face the camera, and moves left hand to the left field of view of the camera (from the user's point of view). A hand icon is shown on the screen to give the user feedback about where the hand position is mapped to (fig. 14b).

Figure 15: Moving the detected hand to the screen’s edge opens up the grab & drop clipboard overlay, which is a global background program displaying the current contents of the clipboard. The grab & drop clipboard currently contains a piece of (rich formatted) text and an image, each with an icon indicating their content type (fig. 15a). As the hand hovers over a slot, it slides in for a preview (fig. 15b).

Figure 16: Moving along the edges changes slots, and the preview changes accordingly (fig 16a and b). The program's response to moving along the edges depends on whether the gesture is an open palm or a fist (not depicted here). The application uses two static semaphoric gestures, and two additional gestures based on the transition between them:

- Open palm: "neutral", or when transitioning from a fist, "drop"
- Fist: "hold", or when transitioning from open palm, "grab"

"Neutral" is used for previewing, "grab" to grab the content, which can then be "held" and "dropped" at the mouse cursor or switched with another slot. The latter is an important feature: not only is allowing the user to rearrange content among the
Fig. 14A: An application supporting Drag & Drop input. 14B: User opens palm to activate multi-clipboard.
5. Portfolio – the exploratory Process

Fig. 15A: Grab & Drop overlay appears. 15B: Previewing a copied piece of rich text.
5. Portfolio – the exploratory Process

Fig. 16a: Navigating Grab & Drop Interface. 16b: Previewing a copied image.
slots more flexible, it would make the metaphor of objects that can be moved around more persistent throughout the behaviour of the application. This makes the whole application behave more consistent to the user, making it easier to anticipate and make sense of. Furthermore, if we would create version of this application with more slots or even stacks, the ability to re-organise becomes even more important.

**Figure 17:** The picture is not what we are looking for right now, so we go back to the text, and drag it into the field (fig 17A). Perhaps a small preview on the bottom-right of the gesture icon would be appropriate, to confirm to the user what content has been grabbed (not depicted). To paste, we open our hand to make a "drop" gesture. It will be inserted at the mouse cursor (fig 17B).

Note that the interacting with this program is very similar to the drag & drop APIs that already exists in most operating systems. In fact, it is entirely possible to build the grab & drop clipboard on top of it (which is one reason why it is suggested to insert dropped items at the mouse cursor instead of the location of the gesturing hand). This has a number of benefits. First of all, the operating system does not need to provide any new special type of support. Second, it would not interfere with the existing clipboard implementation in any way – it would simply be a separate program to and from which items are dragged and dropped with gestures. Third, we get backwards compatibility with all existing software that uses drag & drop functionality and benefit from all of its added features. For example, to get a preview of the to-be-dropped content, as depicted in fig. 17A, would work out of the box with any program that has previews when using drag & drop.

**Figures 18 and 19:** We put the cursor at the beginning of the text with the mouse, touchpad or touch screen (not depicted), while moving our hand to the corner (fig. 18A and B) to grab the picture (fig. 19C) and paste it in the text (fig. 19D). Note that this simultaneous use of the mouse with one hand while the other hand is gesturing is an example of orthogonal bimanual input.

**Figure 20:** Finally, having pasted all items he or she needed, the user puts left hand back on the keyboard and the grab & drop clipboard overlay disappears (fig. 20A), after which he or she can continue editing the document (fig. 20B).
5. Portfolio – the exploratory Process

Fig. 17a: Grabbing and holding copied text. 17b: Dropping held text in an application
5. Portfolio – the exploratory Process

**Fig. 18a:** Moving to copied image. **18b:** Grabbing previewed image
5. Portfolio – the exploratory Process

Fig. 19A: Holding copied image. 19B: Dropping held image in an application
Fig. 20a: Putting down the hand hides the overlay. 20b: Normal interaction with the program resumes
Due to our familiarity with moving around real-world objects and the drag & drop and cut & paste paradigms that are essentially based on the same metaphor, the behaviour of the program is fairly predictable. For example, the storyboard did not show how to copy an object, but it is simple to infer: select items (with mouse or keyboard), hold up open palm in the centre of the screen, grab, move to a slot, drop. One question is whether or not this should result in a copy or a cut – perhaps it is best to leave this as an option the user can switch between. A third option would be to let the drag & drop API handle it by implementing this behaviour as dragging from the environment where the items are selected, and dropping it into the grab & drop clipboard program. In that case, the environment in which the items are selected decides what this behaviour leads to.

All of this together might give the impression that transitioning from existing desktop environments to one with its functionality on top would be almost frictionless. There are however plenty of subtleties to how this tool would interact with existing environments. One is the aforementioned choice of whether putting an item into a storage slot should result in a cut, copy or drag & drop action. Another is how the program should behave if both hands are held up and attempting to interact with items to be copied, rearranged or pasted. In the storyboard above the bimanual input was limited to one hand gesturing, and one hand controlling the mouse, but we could gesture with both hands simultaneously, provided one hand does not block the camera's view of the other (Crunchfish' software, and computer vision in general for that matter, has yet to properly address this issue of “object permanence”). Again, since we base our interface on the metaphor of moving real-world objects around, the equivalent interaction with physical objects is a good starting point for our program's behaviour. Based on that it should be possible to pass a grabbed item from one hand to the other, or to copy/paste/rearrange multiple items at the same time. This interpretation is fairly straightforward. However, what happens if both hands grab the same selected item? In real life, successfully pulling an object in two opposite directions usually tears it apart, destroying it, and we could make the interaction with the virtual environment match this. However, it is unlikely that users desire this functionality, as it would more often than not lead to accidental deletions. Alternatively, we could imagine that pulling an object apart leads to two “clones” of the object. Looking back at our earlier question if grabbing and stashing an object should copy or cut it, this
provides a fourth option: grabbing with one hand could mean cutting it (equivalent to moving a physical object), while the two handed option would function as holding the item “down” with one hand while “tearing away” a copy to the edge.

Similarly, the ability of existing drag & drop APIs to provide previews is thwarted when both hands hold an item to be pasted. Using both hands to drop two items in quick succession could also lead to unexpected behaviour: these would effectively be two drag & drop actions in quick succession. However, dropping the first item would change the state of the program, which might cause it to respond to the dropping of the second item in a different way than the user anticipated.

The ability to use multiple hands suggests being able to have multiple simultaneous pointers, as opposed to the current situation of having one mouse cursor. This is also where we are likely to find interesting design openings for bimanual input. However, most operating systems (let alone applications built on top) currently cannot deal with more than one pointing device at the same time, usually mapping all input to the same mouse pointer (this is for example evident when trying to use a touch-screen and a mouse at the same time). One way to bridge this backwards incompatibility would be to use a scheme similar to how operating systems currently deal with multiple windows: only one window has focus at any given time. The same can be done with pointers: allow for multiple meta-pointers, but give only one focus at any given time. Clicking with any meta-pointer would switch focus – old programs would “perceive” the mouse pointer to instantly move from the location the previously active meta-pointer to the other.

As we can see, the addition of bimanual input and concurrent interactions with environments not designed for it leads to many emergent, complicated scenarios, even in a program with a singular, seemingly simple function. Interestingly we have little problems when performing more or less equivalent kinds of interactions with physical objects in the real-world, partially due to physical restrictions and consistencies not present in our virtual environment. For all of these reasons it was decided that it is better, for now at least, if only the “oldest” hand still being detected is used as

---

6 Having a program or global functionality listen to input when it does not have focus is possible but rarely used – the keyboard short-cuts for screen brightness and speaker volume are two examples.
input for the grab & drop program, and functions as a “meta-pointing device” that exists only within its own application. It would be connected to the rest of the environment through the mouse cursor. However, it seems to indicate another unexplored design space in the venerable desktop paradigm.

5.3.2 Design Proposal: A full-featured drawing application on a small touch screen
Another design direction was to investigate if we could hide large complex interfaces off the edge of the screen, to be pulled in by gestures. This would reserve the limited screen space of the average tablet or phone for the task the program was designed for. It was decided that drawing applications were a good use-case for this. One one hand, the usefulness of a touch screen for drawing is self evident. However, full featured drawing applications such as Adobe Photoshop or the GNU Image Manipulation Program (GIMP) are cluttered with interface elements due to the many advanced and complicated features they provide. Furthermore, on the desktop they often rely on keyboard modifiers and short cuts to complement the input from the mouse or graphics tablet. Because of the limited screen size of a tablet, and the input being limited to touch points on the screen, these advanced and complicated functionalities are difficult to transfer to a tablet from an interaction design point of view.

One potential criticism of using gestures as a solution for this use-case is that tablets are commonly used as media consumption devices on the go, yet bimanual input requires putting the tablet down. Given that one needs the luxury to sit down to do so, one could ask if a desktop computer or laptop would not be a more appropriate solution in that context. While not an invalid question, it is not in any way conducive to the task of exploring the potential design space, and thus will be ignored for this design exercise. Similarly, we ignore the problem of limited computing power and memory, and assume the tablet is capable of running the drawing application and its features. Finally, we ignore the possibility of using peripherals to solve this issue, such as tablet docks with keyboard input, a stylus with buttons or connecting a smart phone with the tablet through Bluetooth, which would give us two separate touch-screens and through that the possibility to distinguish left- and right-handed touch input. These are also interesting design spaces to explore. but beyond the question whether or not they provide an easier solution, they are not relevant to the explora-

---

7 Interestingly, these are examples of bimanual input, although their design rarely appears to be consciously optimised for it, let alone be designed with Guiard’s insights in mind
tion of the particular design space of gesture interfaces. In short, we consciously restrict ourselves to the context of two-handed input with the only possible forms of input being the touch screen and gesture detection through the tablet’s camera, for the purpose of exploring the design space within those restrictions.

Before and partially during the design of the drawing application, existing applications were tried out as a crude form of benchmarking. Five drawing applications for Android systems and two drawing applications iOS systems were tried out in a semi-structured fashion of doodling, looking for interactions where the interface fell short or excelled, and trying to repeat these interactions in the other applications. A stylus was preferred over drawing with fingers. These were my takeaways:

- **Dynamic brush settings — especially size and intensity — are crucial for conveying a sense of control over the medium.** The input for this dynamic behaviour may vary — pen pressure (if the touch-screen supports it), contact surface of the touchpoint size, speed of the brush-stroke. These features somewhat alleviate the pictures-under-glass phenomenon described by Victor, adding an approximation of mapping tactile feedback — pressure, contact surface and force — to input. Regardless of the method, when present it gave a greater sense of control over the medium.

- **Providing a flexible system for customisable brushes makes better use of the digital medium.** Many drawing applications provide a fixed set of categories of different “types” of brushes that try to mimic real-life brush behaviour, while some provided a flexible customisation system. With the latter the brushes can both closely approximate real-life drawing tool behaviour, as well as exploit the digital possibilities of the medium. The former might be more familiar and therefore approachable, but latter system is obviously preferable from a power user standpoint.

- **The shape of the brushes should be decoupled from the drawing tools where they are applied.** These are tools such as freehand drawing, erasing, line- circle- ellipse- and rectangle tools. Applications that do this are more expressive as a result, without adding much complexity.
• **Layers are extremely powerful and remove the need for many other features.** They can work like a surrogate select and copy tool by duplicating a layer and deleting unwanted parts, and with added overlay options allow for non-destructive colouring and desaturating of underlying layers, adding highlights or shadows, etc.

• **Many drawing applications are designed around single-touch input, leading to needless complexity in the interface.** For example: layers typically can be rotated, resized and moved, but as separate operations. In the case of rotation and resizing this often involves movable pivots around which these layers could be rotated or resized respectively. This interaction should be as simple as choosing a single “transform layer” option, then using two fingers to perform these actions simultaneously.

• **UI elements should flow inward from the outside of the screen and go from coarse to detailed settings.** Complex features, if present, would often be found in pop-up menus suddenly appearing elsewhere in the screen. Sliding in from the associated button pressed would be visually and therefore conceptually better connected. It also makes more sense if the conceptual hierarchy goes from coarse to refined concepts, similar to the kinematic chain concept. This essentially describes a pull-down menu, a well-known interface element.

• **Most interfaces are optimised for right-handed users.** For example, drawing applications with layers tend to position these on the right, so that the hand will not obscure the view of the canvas when reordering the layers and the effects can immediately be seen. However, for a left-handed user the view would be obstructed by the hand. Furthermore, left-handed users tend to learn to hold their pen in a different manner than right-handed users when writing to avoid obscuring and smudging letters with their hands. This often translates to holding their stylus differently when drawing. Allowing the positioning of interface elements to be customised, together with sensible defaults for left- and right-handed users, should alleviate most of the issues that follow from these differences.
Furthermore, the following functions and use contexts were identified as first candidates for augmentation by gesture input:

- **UNDO/REDO LAST ACTION.** These are very common operations, and triggering them often comes at the cost of a button being constantly visible or switching from stylus to two-finger swiping. One application used double-tapping in a corner, assigning the left corner for undo and the right for redo. This was a very convenient and fast input method. However, a dynamic gesture like a swipe could be even faster.

- **ZOOM/PAN.** When implemented as a touch-gesture this is usually involves two fingers – making this a touchless gesture allows the touch screen to be used for other multi-touch gestures simultaneously. This opens up the possibility of creating a cooperative bimanual actions. For example, the drawing hand can resize and rotate a layer with a two-finger touch gesture, while the frame-of-reference hand simultaneously zooms in and out and pans around the image. Note how this matches Guiard’s model: zooming and panning is literally about setting a reference frame. Ideally, this interaction is smooth and precise. One self-evident option is to use manipulative gestures, using the horizontal and vertical motion for panning, and distance to the tablet for zoom levels. However, precise depth detection is something that is currently difficult to achieve with a single, built in camera. An acceptable compromise might be using swipe up/swipe down gestures to ease between fixed zoom-levels (25%, 50%, 100% ... 1600%).

- **ACTIVATE COLOUR DROPPER/SET SOURCE PIXEL FOR CLONE STAMP TOOL.** These two functions are found in more advanced drawing applications. While different they have a similar work flow. Both set up a frame of reference for their respective tools (choosing a colour for a brush, setting a source point to be used by the clone stamp tool). Both are often quickly accessed by a modifier key in desktop settings. When only touch input is available this would involve an elaborate mode switch. One option is to use a touchless gesture to signal that the next touch will function as a colour dropper or set the source pixel, after which the use of the previously selected tools can be resumed.
• **REVEALING AND HIDING MENUS.** The menus should slide in from the sides of the screen, making gestures involving dynamic motion mapped to this behaviour a logical option. Swiping gestures are one possibility, but if those are mapped to undo/redo, then a grabbing and pulling gesture similar to the one used in the grab & drop clipboard concept is another option.

• **MANIPULATING DYNAMIC BRUSH SETTINGS.** One major benefit here would be the ability to change the settings with the frame of reference hand while the line is being drawn by the drawing hand. Similar forms of bimanual input have been explored in drawing applications many times before, for example by (Owen et al., 1998). Another design opening related to this idea would be *retroactively* applying the new brush settings to the entire current brush stroke, or last brush stroke if not currently drawing. While technically not complicated to implement, this second idea is currently not exploited by any raster-drawing applications that we know of.

Using the frame-of-reference hand for gesture input reduces function overloading for touch input of the drawing hand. There are still mode switches between drawing and selecting, but all frame-of-reference-related tasks are delegated to the frame-of-
reference hand. Furthermore, this enables cooperative bimanual actions, with the
drawing hand creating the brush strokes while the frame-of-reference hand changes
the brush settings or allows for zooming and panning. This follows Guiard’s principle
of the frame-of-reference hand having precedence: the settings of the brush set up
the reference frame for what the brush strokes will end up looking like.

One problem encountered is finding a large enough vocabulary of gestures to ap-
ply to all of these forms of input – especially if we want to give the user control over all
dynamic brush settings. Furthermore, the input for zooming, panning and dynamic
brush settings is not a matter of turning a function on or off, but needs some form of
variable input. This narrows down the design space for these gestures. A similar
problem is finding the appropriate gesture per function, although research indicates
that giving the user the freedom to define his or her own gesture may be preferable
(Nacenta et al., 2013). While Crunchfish’ technology requires their gesture detector to
be trained, it is still technically possible to let the user remap these predefined ges-
tures to different functions, like customisable key-bindings in desktop programs.

A mock-up was designed base on these ideas, with the lay-out organised with a
right-handed user in mind. Three menu-groups were identified: brushes and associ-
ated tools, colours, and layers and associated tools.

One early design decision to make in regard to these menu groups was whether
the gestures for pulling them in should do so from the sides, the corners or both.
Corners were chosen, as this would make the gesture motion diagonal relative to the
camera. This would allow for slightly more detectable distance to be crossed and in all
likelihood make the detection more reliable.

Another question was whether a menu should remain visible until dismissed, or
immediately disappear if the gesturing hand would “let go.” With a few exceptions
the latter behaviour, which was dubbed “peek mode”, was chosen for the general case.
The time and effort spent on something different than the main the task at hand –
drawing – should be minimal. By arranging the menus properly, we can ensure that
the most common actions that would require menu access require just one tap once
the menu is visible. Requiring a tap to dismiss the menu would double the amount of
work then! The exception to this behaviour would be entering detailed sub-menus,
which is a more involved action that likely requires more attention from the user. In
this context the menu should stay up until the action in the sub-menu is finished. With that in mind it is preferable for the user to be able to safely put down the frame-of-reference hand, to prevent it from getting tired.

5.3.2.1 Brushes and brush tools

To limit the scope of the mock-up somewhat, the chosen brush tools were limited to drawing, erasing and clone stamp (fig. 21). Note how switching between drawing modes (draw, erase, stamp) or pre-defined brushes requires only one tap.

One feature of desktop drawing suites is the ability to choose between advanced drawing modes for the brush tools, such as colour burning/dodging, or only colourising the underlying image without affecting light levels. Similarly, one can let the erase tool apply to the current layer, all visible layers or all layers, and have the source area of the clone stamp tool look only at one layer, all visible layers or all layers. To include these options without having them clutter the space of the screen, these can be accessed by tapping and holding the associated tool, then dragging and “pulling out the menu” of this tool (fig. 22). One then chooses an option by removing the stylus from the screen. This drag-to-pull metaphor is an example of starting with coarse-grained, more common options (tap to switch between drawing or erasing) and flowing inward into the screen into more detailed settings (drag to choose specific drawing or erasing mode). It is also essentially a variant of the venerable pull-down menu of traditional desktop GUIs. Being similar to this old menu, it was expected to be familiar and therefore relatively intuitive. It is therefore used in other places in the interface.

All brushes are procedurally generated, and can be pulled out to review their detailed settings (fig. 23). The precise layout of these settings was not drawn, but among the possible brush settings for individual brushes would be:

- Bristle shape, density and size
- Intensity (with dynamic modifier)
- Angle (with dynamic modifier)
- Spacing (with dynamic modifier)
- Jitter (with dynamic modifier)
5. Portfolio – the exploratory Process

Fig. 22: Pulling out a detailed settings menu

Fig. 23: A very rough sketch of pulling out the detailed brush settings menu – creating a specific layout for all possible settings was skipped for this mock-up
5. Portfolio – the exploratory Process

A button to duplicate or delete a brush should also be present, as should the possibility to remap gestures to manipulate these brush settings.

Accessing the detailed brush settings and tweaking them to create specific brush behaviour is a more involved process, which the user is more likely to spend some time on. This then would be an example of where a menu should be visible until explicitly dismissed once pulled out. This could be done by either tapping outside the detailed menu, or with a dismissive swipe gesture.

5.3.2.2 Colour panel

As changing colours is an extremely common operation, it was decided to put it into its own menu group for fast access (fig. 24). The colour panel should contain a colour wheel, an opacity setting (not depicted) and a palette for storing colours. Dragging one palette item on another should duplicate that colour (not depicted), to allow for one colour to function as a template and make it easy to create subtle varieties on it.

Because it is likely that brush shape and colour are changed in tandem, these menu groups were put in opposite corners to make the gesture required for switching between them as distinct as possible (figures 25 and 26).
Fig. 25: Switching brush shape and colour without leaving “peek mode”
5.3.2.3 Layers, layer options, transformations and filters.

These are all operations that apply to an entire layer, such as inverting its colours, resizing and rotating, applying a blur filter. It therefore makes sense to group them all together (fig. 26). "Pull-menu" options should be provided for layer blend modes (fig. 27a), delete layer (delete visible, delete hidden), duplicate layer (new empty layer, duplicate merged visible), and merge buttons (merge visible below, merge all visible). As this example shows a right-handed layout, the elements are positioned along the bottom and right sides, to avoid the drawing hand from obscuring the view. When an option is selected, such as layer mode, all other interface elements should be hidden so that as much of the screen is dedicated to a live preview of what the consequences of the actions will be (fig. 27b). This should also apply to any of the (not depicted) layer manipulation and filter options. These could be filters such as blur or sharpen, and tools that change the entire layers such as a curves tool, the manipulation of hue, saturation and value, and a channel mixer.
While the mock-up is a little rough around the edges, it hopefully gives a decent impression of the suggested interaction flow. What sadly is impossible to convey in static images is the possibility for using cooperative bimanual input in a number of situations within this drawing application:

- Changing dynamic brush settings with the frame-of-reference hand while the drawing hand is creating a brush-stroke. Certain colour settings (lightness, saturation, opacity) are also candidates for retro-active manipulations.

- Zooming in and out and panning around the image with the frame-of-reference hand while performing layer manipulations with the drawing hand, such as layer reordering or layer resizing and rotation.

These combinations highlight how orthogonal but complementary actions can be combined into cooperative bimanual actions, to give the user a stronger sense of control, while making the interface less confusing and faster to use. Because individual hands and forms of input (touch, gesture) are less overloaded in this interface, it is less likely that two actions get confused. Furthermore, when the actions do not have to be performed in sequence this can be more than twice as fast, as the sequential manipulations would likely require a number of back-and-forth iterations to get the desired output.
6 THE EMERGING DESIGN SPACE OF GESTURE INTERFACES

Looking back at all the different design prototypes and their associated discussions, in other words at the engagements with the medium of gestures in an interface context, we can observe more general takeaways and general directions we might want to investigate further.

6.1 GESTURES IN THE CONTEXT OF DESIGNING BIMANUAL INTERFACES

One design space that gesture detection opens up for phones and tablets is allowing for bimanual interfaces. In existing research it is suggested that this can greatly improve the fluidity of interaction and reduce cognitive load on the user (Leganchuk et al., 1998). This would be achieved by dividing tasks in accordance to whether the action sets up a frame of reference, or is an action that needs to be performed within this frame of reference. Looking at drawing applications as an example, one such combination of tasks is pulling up various menus with one hand (frame-of-reference) and selecting an item in such a menu with the other. Another is zooming/panning (frame of reference), while simultaneously performing manipulations in the current view, such as layer reordering or resizing and rotating. A third is changing brush settings and colour (frame of reference) while performing a brush stroke. What should be noted is that all of these are tasks that already exist in isolation in current drawing applications. The idea that interface elements should be structured hierarchically from coarse-grained functionality to fine-grained aligns with Guiard’s kinematic chain. Identifying frame of reference-type hierarchies may be a general way to redesign cooperative bimanual interfaces within existing complex interfaces.

6.2 GESTURES IN THE CONTEXT OF SOCIAL INTERACTION AND ATTENTION

The pervasiveness of smart phones and tablets with their “always online” has implications regarding attention not being available to other tasks. The vulnerability of social interactions discussed in the knowledge database design proposal is one example. However, this does not have to be inherent to the interaction with a device – it is the fact that most of the applications on these devices are designed around a one-on-one interaction with a single user that promotes this behaviour. Design can and should do more to facilitate local social interaction, and the publicly readable nature of gestures, as well as the capability to detect gestures from multiple users simultaneously, make
6. The emerging design space of gesture interfaces

gesture detection an enabling technology for such interfaces. One related use context is that of collaborative work platforms. Given the performative strength of gestures we have reasons to believe gestures may work very well in the context of collaborative work platforms, but technology does not exist in a vacuum, nor can we as designers properly engage with the material when the actual context is absent due to other technical limitations (Clearboard) or being a highly niche situation (Line Control Rooms). Therefore the direction of gestures in collaborative work settings remains rather unexplored.

One way to address the problem of attention being drawn to the screen and away from the surroundings is by removing the need to interact with the screen in the first place. Gesture detection is based on camera sensing which – similar to voice activation and other forms of sound detection – extends the interaction zone of the phone or tablet beyond the touch surface, in this case to the field of view of the camera(s). As a result it removes need for the users to be focused on the screen, as the Blind Cyclops game demonstrates. The opposite direction is also possible: in the case of the Minotaur game the camera and screen of the phone become an extension of the senses, while the fake HUD purposely adds layer of mediation.

6.3 Gestures in the context of embodiment

Gesture detection allows for very embodied forms of interaction, either “outwards” by extending the senses of the surrounding world, or “inwards” by extending the embodied experience into the virtual environment of the computer. For example, by allowing us to “reach into” a virtual space, like in the grab & drop clipboard. Recall how part of the design idea of grab & drop clipboard was to rely on spatial memory. This may be a too simplistic take on the matter. Compare the following two question commonly asked when looking for a lost object:

- “where did you last see it?”
- “where did you last put it?”

The first question is one that relies purely on the visual aspect of spatial memory. The latter – which applies to the grab & drop interface – implies personal agency, and physical interaction with the object. It is a more embodied question, and when answering it the user can use both visual and embodied spatial memory. In other words: the grab & drop interface allows for a simple form of embodied cognition to take place.
This points at a general movement that seems to take place in interface design in the direction of more embodied forms of interaction. Compare the command line, relying text input on a keyboard, to the graphical user interface using a mouse pointer to click on icons. The latter is a significantly more embodied experience, through usage of manipulative gestures. Assuming this trend continues, bimanual gesture interfaces may allow for taking yet another step in this direction.

6.4 Gestures in the context of human ex-centricity

Recall that according Kockelkoren’s essay (implicitly) argued that the search for “natural” user interfaces begins with a flawed premise, because there is no static natural foundation to get closer to. Instead, we fundamentally are in a permanent state of “oscillation between decentring and recentring.” While it was not attempted to validate this philosophy, the designs were made taking it as a starting point. For example, “naturally artificial human ex-centricity” is manifested quite literally in the game designs: they use the phone to mediate an artificial extension of the senses, and in doing so intend to throw us off balance as part of their core gameplay.

Moreover, this philosophy promoted a general scepticism to the “purpose” of a phone. Rejecting the idea of a static ideal to be discovered, and instead favouring constant reinvention of what is considered “natural”, appears to have resulted in a similar rejection of functional fixedness in the design process. We can see this across the whole project. First, there is the early analysis of implicit design assumptions, which attempts to break down the traditional roles of mobile devices. The game designs’ use of mobile phones as wearable artificial senses is an example of breaking out of these traditional roles. To some degree it can even be observed in the proposed drawing application, which rejects the assumption that tablets and mobile phones are unsuitable for tasks beyond a certain level of complexity. Therefore, regardless of the validity of this philosophy, it seems to encourage a fruitful attitude when exploring new design spaces. It promotes analysing how we currently view things to “be”, in order to break free of this view, allowing us to instead explore what they can become.

---

8 The cognitive bias to view and use an object only in the way it is traditionally used.
7 Final Reflections and Takeaways

With this portfolio I hope to have shown a few viable directions that we could explore in our attempts at taming gesture interfaces. What I have purposely avoided throughout the entire design process is the notion that one interface is more “intuitive” or “natural” than the other. I must admit to being personally biased against the belief that gesture interfaces will provide a more “natural” form of interaction, for a multitude of personal reasons: I am red-green colour blind, reducing many “intuitive” colour schemes to “barely readable.” I am deaf in my left ear, diminishing the experience in many multi-media entertainment environments. Finally, I am left-handed, so many tools made to be “easier” to handle are instead clumsy, painful and sometimes dangerous for me to use. Minor physical inhibitions thus have consequences in my usage of many artefacts, and point to the often implicit assumptions made in their designs. These are constant personal reminders of the importance of paying attention to the embodied aspects of interaction design, which may explain my interest in phenomenological philosophy in this context. In my opinion the claim that gestures are more “natural” to use presumes too much about the user – for example, we have completely ignored if the gesture detection software would still work if the user would be missing a finger or has an extra one. Even if it does, will the interface work as expected or does the design somehow depend on possessing exactly five fingers?

Contrary to my scepticism regarding “natural” interfaces, I do think gesture interfaces will play a role in the “permanent oscillation between decentring and recentring.” To develop gestures as their own medium requires letting go of traditional assumptions – a “decentring” from the “temporary equilibrium”. Doing so will, among other things, enable forms embodied interaction that can involve embodied cognition, allow for artificial extensions of the senses into both the physical and the virtual world, provide a method for publicly readable, multi-user input, and a way of taming input complexity through asymmetrical cooperative bimanual interfaces.

Kockelkoren claims that mediatory technologies are the engines of change in the constant reinvention of our “natural artificiality”. How the further exploration of the gesture interface design space will affect us will show if that is actually so.
8 Bibliography


