ORIGINAL ARTICLE

Consistent, Continuous, and Customizable Mid-Air Gesture Interaction for Browsing Multimedia Objects on Large Displays

Arthur Sluÿters[a], Quentin Sellier[a], Jean Vanderdonckt[a], Vik Parthiban[b], and Pattie Maes[b]

[a]Université catholique de Louvain (UCLouvain), Louvain Research Institute of Organizations and Management (LouRIM), Louvain-la-Neuve, Belgium;
[b]Massachusetts Institute of Technology, MIT Media Lab, Cambridge, United States of America

ARTICLE HISTORY
Compiled July 29, 2022

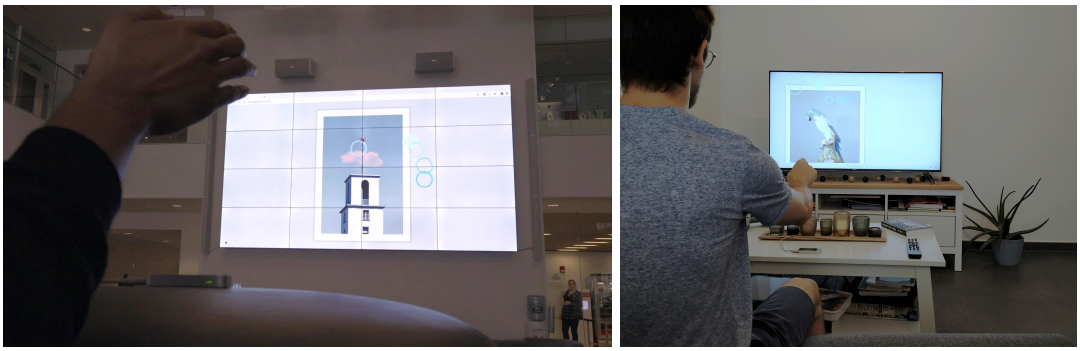
ABSTRACT
Browsing multimedia objects, such as photos, videos, documents, and maps represents a frequent activity in a context of use where an end-user interacts on a large vertical display close to bystanders, such as a meeting in a corporate environment or a family display at home. In these contexts, mid-air gesture interaction is suitable to a large variety of end-users, provided that gestures are consistently mapped to similar functions across media types. We present Lui (Large User Interface), a ready-to-deploy and to-use application for browsing multimedia objects by consistent mid-air gesture interaction on a large display that is customizable by mapping new gesture classes to functions in real-time. The method followed to design the gesture interaction and to develop the application consists of four stages: (1) a contextual gesture elicitation study (23 participants × 18 referents = 414 proposed gestures) is conducted with the various media types to determine a consensus set satisfying consistency, (2) the continuous integration of this consensus set with gesture recognizers into a pipeline software architecture, (3) a comparative testing of these recognizers on the consensus set to configure the pipeline with the most efficient ones, and (4) an evaluation of the interface regarding its global quality and specific to the implemented gestures.

KEYWORDS
Consistency; Gesture-based interfaces; Gesture recognition; Large display; Leap Motion Controller; Mid-air gestures; Multimedia contents

1. Introduction

Large displays are becoming increasingly popular in recent years (Ardito et al., 2015). They typically represent a single contiguous space, composed of a single display unit or multiple tiled units forming a larger display that serves for both input and output (Andrews et al., 2011). Their large size and high resolution enable to display a large amount of information with a high level of detail in many disciplines (Rajabiyazdi et al., 2015), but usually require users to stand further away from the display to grasp all of the displayed contents (Vogel & Balakrishnan, 2005). Using keyboard and mouse

CONTACT Arthur Sluÿters. Email: arthur.sluyters@uclouvain.be



(a) Meeting in a corporate environment. (b) Family display at home.

Figure 1.: Two Lui contexts of use: a meeting in a corporate environment (a) and a family living room (b).

input is difficult on these large displays because small pointer movements often do not scale well with the display size and individual elements on screen.

Consequently, mid-air gestures have been proposed as an alternative to traditional input methods (Vatavu & Zaiti, 2014). They can efficiently leverage the large display area (Nancel et al., 2011) and the space available in front of the display to augment its capabilities (Yoo et al., 2015). However, mid-air gestural interfaces are not very well known by end-users, they have little or no experience with mid-air gestures as they have with touch gestures (Morris et al., 2014), and transitioning from 2D touch gestures to 3D mid-air gestures is not straightforward (Boulabiar et al., 2014). The large degree of freedom allowed by mid-air gestures results in a wide range of possible gesture classes produced with varying articulations (Vatavu, 2013). Multimedia is a media comprised of numerous forms of information contents and information processing, *e.g.*, text, audio, graphics, animation, and video, which can be accessed, played, replayed, and displayed on many devices (Steinmetz & Nahrstedt, 2004). In this paper, we are interested in browsing multimedia objects such as photos, videos, documents, and maps, as they are among the most frequently manipulated contents (Drossis et al., 2013) in multiple contexts of use (Calvary et al., 2003). For example, people in a corporate meeting exchange and sort documents (Fig. 1a), family members collect and organize their pictures into albums (Fig. 1b), movie lovers structure their film collections, and trip organizers examine geographical maps to decide a trip plan.

Designing an intuitive mid-air gestural interface for browsing these multimedia objects is a complex process involving designers and end users, *e.g.*, by participatory design, by conducting a Gesture Elicitation Study (GES) (Wobbrock et al., 2009). This interface should ensure some consistency across media types to help end-users who learned gestures for one media type to reuse them for another type. This knowledge transfer is particularly suitable for gestural interfaces as they are challenging to learn (Fruchard et al., 2018), to produce (Vatavu, 2013), and to remember (Nacenta et al., 2013).

Using the same mid-air gestures on different media types poses consistency challenges (Dingler et al., 2018). Each media possesses its own set of functions, some being generic across all types (*e.g.*, display a media fullscreen or not) and some being specific to a single type (*e.g.*, place a marker on a map). As media type diversity increases, using a mid-air gesture for one media type should enable end-users to transfer this gestural knowledge to other types when appropriate. While many GES have been conducted (Villarreal-Narvaez et al., 2020) for a single context of use at a time (*e.g.*, for a single media with a single device), very few studies have investigated gestures

across media boundaries. Therefore, in this paper, we aimed at developing a better understanding of the elicitation of a mid-air gesture set that is **consistent** across media types. A second problem arises when transitioning from gesture elicitation to recognition: gestures resulting from elicitation are very rarely, if ever, reused for recognition as these two processes are kept separate (Villarreal-Narvaez et al., 2020). In this paper, the output of the gesture elicitation study serves as input to the gesture recognition via a **continuous** process. When some end-users still prefer to rely on their own gestures, which can be different from the elicited gestures, the interface should remain **customizable** to enable end-users to add their own gestures (Coyette et al., 2007). Gesture recognition and their customazibility should also remain continuous (Kohlsdorf & Starner, 2013; Caputo et al., 2019).

To address these challenges, we present a Large User Interface (Lui) for browsing objects of various multimedia types, *i.e.*, photos, videos, documents, and maps, on large displays as a means to investigate consistent, continuous, and customizable mid-air gestures, with the following contributions:

(1) A method continuously integrating a consistent gesture set (resulting from a contextual gesture elicitation) with its recognizers into a pipeline software architecture, so that gestures can be recognized accurately and continuously.
(2) An integration of these gestures and their recognizers into Multimedia User Interface (Lui), a ready-to-deploy, ready-to-use, and customizable interactive application for browsing multimedia objects on large displays.

This paper is structured as follows: Section 2 reviews existing work in gestural User Interfaces (UIs). Section 3 overviews the Lui application, its concepts, software architecture, design requirements, and summarizes the differences between its three iterations. Section 4 details the stages of the development method used for Lui, including the design of a consistent gesture set, the implementation, and the configuration of a gesture recognition pipeline. Section 5 evaluates the interface with respect to its overall quality and the memorability and recognition of the gestures implemented. Section 6 discusses the Lui benefits and limitations and its development method. Finally, Section 7 concludes this paper.

Open Science. The Lui open source code is publicly available on GitHub at `https://github.com/sluyters/QuantumLeap`. Our website at `https://sites.uclouvain.be/ingenious/projects/quantumleap/` provides the reader with more resources, including the datasets used in the testing conducted in Section 4.3 and the spreadsheets of our GES. Demonstration videos of the Lui application are accessible on YouTube at the playlist `https://youtu.be/b0VRvXWFoEs`.

2. Related Work

This paper is at the crossroads of mid-air gesture interaction on large displays and gesture elicitation studies, particularly those focusing on multimedia, which we discuss in the next subsections.

2.1. Mid-air Gestural Interaction on Large Displays

In the field of 3D gesture interaction (LaViola, 2013), hand gesture interaction (see (Cheng et al., 2016) for a survey) is particularly attractive for mid-air interaction: the hand is one of the most mobile human limbs, many efficient gesture recognizers

exist in Machine Learning (ML) (Yasen & Jusoh, 2019), in particular for Template Matching (Brunelli, 2009). Techniques for mid-air interaction with large screens have been studied in the literature (Koutsabasis & Vogiatzidakis, 2019), which reveal its shortcomings, such as arm fatigue (Gupta et al., 2017), need for short, pleasing, and easy to recognize gestures (Kohlsdorf & Starner, 2013), and lack of consistency (Gronewald et al., 2016). Moreover, these techniques often focus on a few actions only, such as point-and-click or pan-and-zoom (Nacenta et al., 2013), or on a limited set of gestures (Gronewald et al., 2016). Integrating these techniques may prove challenging.

For example, (Vogel & Balakrishnan, 2005) compared different techniques for free-hand pointing and selecting a target on a large display: two clicking techniques (airtap: tap vertically with the index finger, like clicking with a mouse, and thumb trigger: move the thumb toward the index) and three pointing techniques (ray casting, relative pointing with clutching, and hybrid ray-to-relative pointing). They implemented auditory and visual feedback to compensate for the lack of tactile feedback inherent to mid-air gestural interfaces. Although clicking techniques were equally performant and accurate, airtap induced less fatigue than thumb trigger. For the pointing, participants preferred "relative pointing with clutching", as it felt faster and easier to perform.

(Yoo et al., 2015) compared two techniques for interacting with large displays: point and dwell and a combination of push and grab-and-pull. In the context of navigating Twitter feeds on a public display using a Microsoft Kinect for gesture recognition, the combination of push and grab-and-pull gestures was deemed more fun, faster to perform, and less tiring than point-and-dwell.

Several factors influence the quality of mid-air gestures on large displays, such as position and direction (Fruchard et al., 2018), sideways hand extension (Koutsabasis & Domouzis, 2016), dimension and bit cardinality (Vatavu, 2013), and continuity (Kohlsdorf & Starner, 2013). Beyond these factors, (Nancel et al., 2011) identified three factors for designing mid-air pan and zoom navigation on large displays: uni- *vs.* bimanual interaction, linear *vs.* circular motion, and level of guidance (from high guidance with a 1-dimension motion on an input device, to low guidance with a free 3D motion in mid-air). They studied the impact of these factors on users' performance and subjective preference by comparing 12 mid-air pan-and-zoom techniques with various factor combinations. Despite its appeal, free mid-air motion was less efficient and more tiring than other techniques. Overall, users preferred linear motion and bimanual interaction over other techniques.

More specifically for mid-air interaction with multimedia objects, less work can be found and usually focus on one media type or two at a time but not as a whole: images (Koutsabasis & Domouzis, 2016), videos (Zigelbaum et al., 2010), virtual objects (Caputo & Giachetti, 2015), 3D travel (Ortega et al., 2017), images and videos (Drossis et al., 2013). For example, (Ackad et al., 2015) designed and evaluated a "media ribbon", a large public display that shows information about events for theaters and concerts. Designing an intuitive and easy-to-learn gestural interface is more critical for such displays, as people are less inclined to take time to learn how to interact. They came up with four gestures (swipe left/right and move the hand up/down then hold briefly) that could trigger five different commands. To evaluate the interface, they let passers-by interact freely with the large display and interviewed them afterward. They concluded to keep gesture sets small and easy-to-learn, *e.g.*, by always showing the most important gestures on the display and by providing context-dependent tutorials for additional gestures.

(Vlaming et al., 2008) used a modified Wiimote and an infrared light source to track four retro-reflective points (two points per hand, on the index and thumb). Their

system allowed users to interact with documents, videos, and pictures on a large screen through a combination of pointing and "pinch" gestures. Media-dependent widgets on the screen could trigger specific actions (*e.g.*, play/pause a video or go the next slide) when performing a pinch gesture while pointing at them. Content could be rotated and translated using either one (Rotate N' Translate) or two hands (two-point rotation). The two-point rotation technique allowed users to resize the selected element. In a testing with five users, the authors noticed that the system could lead to arm fatigue when used for extended periods and that some participants had difficulty realizing that two hands were necessary to perform some actions.

Zigelbaum *et al.*'s G-stalt (Zigelbaum et al., 2010) is a chirocentric gestural interface for manipulating videos in a 3D graphical space. It featured an extensive gesture set of 21 gestures based on Oblong Industries' g-speak and relied on a Vicon motion capture system to track passive IR retro-reflective dots placed on the hands for gesture recognition, which created 3D point clouds (Cook et al., 2016). The actual interface consisted of a cubical arrangement of media such as photos and videos, which the user could sort through, play, and reorganize the structure into meaningful arrangement. This cube of media could be further rotated and zoomed using a set of hand motions. The gesture set involved actions such as two-handed pinch, telekinetic actions, stop all, lock, and unlock. While providing sub-millimeter precision, the system is expensive and must be carefully set up. The authors recommended designing gestural interfaces that are neither too complex nor too simple and taking advantage of the hands' capabilities while keeping gestures easy to learn and perform.

SpaceTop (Lee et al., 2013) is an integrated 2D and 3D spatial environment with a see-through LCD display to better support spatial memory. The system enables a more immersive experience for the end-user, providing the ability to do document editing or 3D modeling without being restricted to 2D. A first MS Kinect camera pointed towards the user's head to enable motion parallax and a second MS Kinect pointed down at the user's hands to detect position and pinch gestures. One key interaction method revolves around the position of the user's hands. When the user lifted her hands, the 2D display would fade out or slide up to reveal a 3D interface. Conversely, when the user placed her hands on the surface, the display would revert back to 2D. Similarly in Lui, the interface recognizes the hands when they are lifted about the sensor and remains locked when the user removes them. End-users felt comfortable sifting through a pile of documents with one hand while the other was focused on the main task. This interface confuses some users when switching between the 2D and 3D modes.

(Jakobsen et al., 2015) compared touch and mid-air gestures for selecting objects on large displays. In the touch interface, an object could be selected by touching it on the screen. In the mid-air interface, a cursor was displayed on the screen with ray casting and an object could be selected by performing a "thumb trigger" gesture (*i.e.*, touching the middle finger with the thumb) while pointing at the object. In a first experiment, they compared the accuracy, time, and user preference between touch and mid-air gestures for selecting targets of various sizes separated by various distances on a large screen. They showed that touch was usually faster and more accurate than mid-air gestures, but that this advantage decreased with the distance between targets, as mid-air gestures allowed users to see and interact with targets anywhere on the screen with minimal motion. The poor performance of mid-air gestures could be explained in part by an imperfect implementation of the "thumb trigger" gesture. In a second experiment, they showed that users mostly favored mid-air gestures when they were regularly forced to move away from the screen (*e.g.*, to interact with something else), despite the loss in accuracy incurred.

(Siddhpuria et al., 2017) studied the possibility of using "at-your-side" gestures (*i.e.*, gestures performed with the arm at rest along the side of the body) to control (semi-)public displays and applications while minimizing the effort associated with typical mid-air gestural interaction. They conducted a gesture elicitation study to determine participants' preferred gestures for a set of 34 tasks. The authors advised considering user-specific gesture sets as, while the overall inter-user agreement is low, individual recall is high. They also observed that users tended to combine multiple symbols into a single gesture that conveys a different meaning. To evaluate the feasibility of "at-your-side" gestures, they implemented a working prototype using a smartwatch, a threshold-based segmentation technique, and a template-matching recognizer. They reported that the system was reasonably accurate (>80% accuracy) and considered as low-effort and low embarrassment by users.

The closest system to our Lui is MAGIC (Drossis et al., 2013), a system for browsing a collection of pictures and videos on a large display in public spaces with mid-air gestures captured by a Microsoft Kinect. MAGIC is similar to Lui in terms of objectives, *i.e.*, browsing multimedia objects, but in a different setup (a Leap Motion Controller instead of a Kinect) with consistency and customizability as added requirements. Reported usability issues include: the distance may affect the recognition accuracy, such as the Swipe gestures being incorrectly recognized or tap to click not accurate enough. A subsequent version replaced these gestures by a combination of buttons and point-and-dwell. In addition, the set of functions is quite limited: scroll a gallery vertically/horizontally, show details of a multimedia object, change caption language, switch to an adjacent element, and magnify an object.

2.2. Gesture Elicitation Studies

(Wobbrock et al., 2009) invented a participatory design method called "gesture elicitation study" (GES) to understand, collect, explore, and analyze users preferences for gestures suitable to a particular context of use (Calvary et al., 2003), which is made up of a user and related tasks, a device or computing platform, and an environment. The GES outcome consists in a characterization of users' gesture input behavior in terms of user-defined gestures (Grijincu et al., 2014) with valuable information for designers, practitioners, and end users regarding the consensus levels between participants, computed as agreement or co-agreement scores (Vatavu & Zaiti, 2014), or more recently agreement rates (Vatavu & Wobbrock, 2015). This method has been widely applied to a large array of individual contexts of use (see (Villarreal-Narvaez et al., 2020) for a systematic literature review), such as in Augmented Reality (Piumsomboon et al., 2003), smartphones (Ruiz et al., 2011), and holograms (Pham et al., 2018).

The vast majority of these GES target a single context of use, which means a single category of users issuing gestures with a dedicated device in a given environment, thus posing the question of transferrability to other, possibly uncovered, contexts, and their applicability. Much less GES have carried out on eliciting gestures in multiple contexts simultaneously, which induces challenges the consistency of gestures across these contexts. (Anthony et al., 2013) already detected that stroke gestures issued by a stylus *vs.* finger are inconsistent. (Gronewald et al., 2016) performed also a systematic literature review of mid-air gestures to observe that gesture sets were largely inconsistent with each other and that no holistic study exists to address this problem. This is an issue for our Lui.

(Vatavu & Zaiti, 2014) studied users' mid-air gesture preferences for interacting with

a smart television with a Leap Motion Controller. While the overall agreement was low, 82% of the participants favored gestures instead of the TV-remote. The agreement rate seems to correlate with the recall rate, *i.e.*, a higher agreement usually means that participants recall their gesture propositions better.

As this study was *monomodal*, (Morris, 2012) investigated a *multimodal* version for this TV context. Participants proposed speech commands, gestures, or a combination of both for 15 web browsing functions. Participants could also give multiple proposals for the same referent. Two new measures evaluate consensus in this case: max consensus and consensus-distinct ratio.

(Dong et al., 2015) conducted a GES for common navigation actions, such as snap fingers to open contents, push to enter a menu, move the hand in any direction for channel surfing, and move a flat hand up/down to increase/decrease a volume.

(Wittorf & Jakobsen, 2016) conducted a GES to elicit mid-air gestures for interacting with elements on wall-displays. They noted that users' hand poses were often irrelevant to the meaning of the proposed gestures (*e.g.*, a "swipe" gesture can be performed with any hand pose). Instead, the meaning was conveyed through the direction or expression of the gesture. When referents resembled actions associated with touch interactions, the proposed gestures were often larger-scale versions of common touch gestures. Users tended to favor gestures resembling the physical manipulation of objects. Overall, compared to other studies on smaller displays, the participants of this study proposed larger and more physically-based gestures.

(Ortega et al., 2017) conducted a GES for eliciting mid-air gestures for 3D navigation. The Swipe emerged as the preferred gesture for translation (*e.g.*, swipe up/down/forward/backward for moving to the corresponding direction), the flickwrist and the rotate gestures for rotation, and the pinch in/out gesture for scaling. While these gestures are consistent with respect to the rotation-scaling-translation invariance properties, they are mainly intended for navigating within a 3D scene.

Consistency has been initially addressed in three studies. (Dingler et al., 2018) designed a gesture set for reading via RSVP (Rapid Serial Visual Presentation) that ensures consistency across three devices, *i.e.*, a smartwatch, a smartphone, and smart glasses. A first GES collected gesture proposals for all three device types at once and considered the gesture proposals for all devices when constructing the final gesture set. They defined two new measures: the *transferability score* (*i.e.*, how well a set of commands learned on one device can be ported to another) and the *consistency score* (*i.e.*, the mean of all transferability scores between all devices). A second experiment with 18 participants validated the transferability of their gesture sets across devices. More recently, (Vogiatzidakis & Koutsabasis, 2019) computed a consistency rate $CR(C)$ by computing the agreement for one user performing one our of 14 commands (*e.g.*, turn on/off, up/down/next/previous on 7 home devices (*i.e.*, a smart television, speakers, a video player, audio player, an air conditioner, lights, and blinds). For this purpose, they compute the agreement rate for a single command for a single user, but for different devices. This is similar in principle to our case where the same command could be used on different media types, but always on the same device. In the same home context of use, (Vogiatzidakis & Koutsabasis, 2020) introduced a consistency routine process that checks gesture proposals for consistency before incorporating them into a real system.

In conclusion, apart from the three last studies, we are not aware of any work investigating gesture consistency, either in general or for mid-air interaction, across different types of media objects. Some papers study users' preferred gestures either for a particular media type or for a well-defined set of actions, but independently of media types or without considering inter-media consistency. When consistency is addressed,

Function	Media type			
	Photos	Videos	Documents	Maps
Fullscreen on/off	●	●	●	●
Like/dislike	●	●	●	●
Previous/next	●	●	●	○
Zoom in/out	●	○	●	●
Rotate	●	○	●	●
Pan	●	○	●	●
Change brightness	●	○	○	○
Change contrast	●	○	○	○
Play/pause	○	●	○	○
Show/hide subtitles	○	●	○	○
Volume up/down	○	●	○	○
Fast-forward/backward	○	●	○	○
Highlight text	○	○	●	○
Place marker	○	○	○	●
Change layer	○	○	○	●

Table 1.: List of functions for each media type.

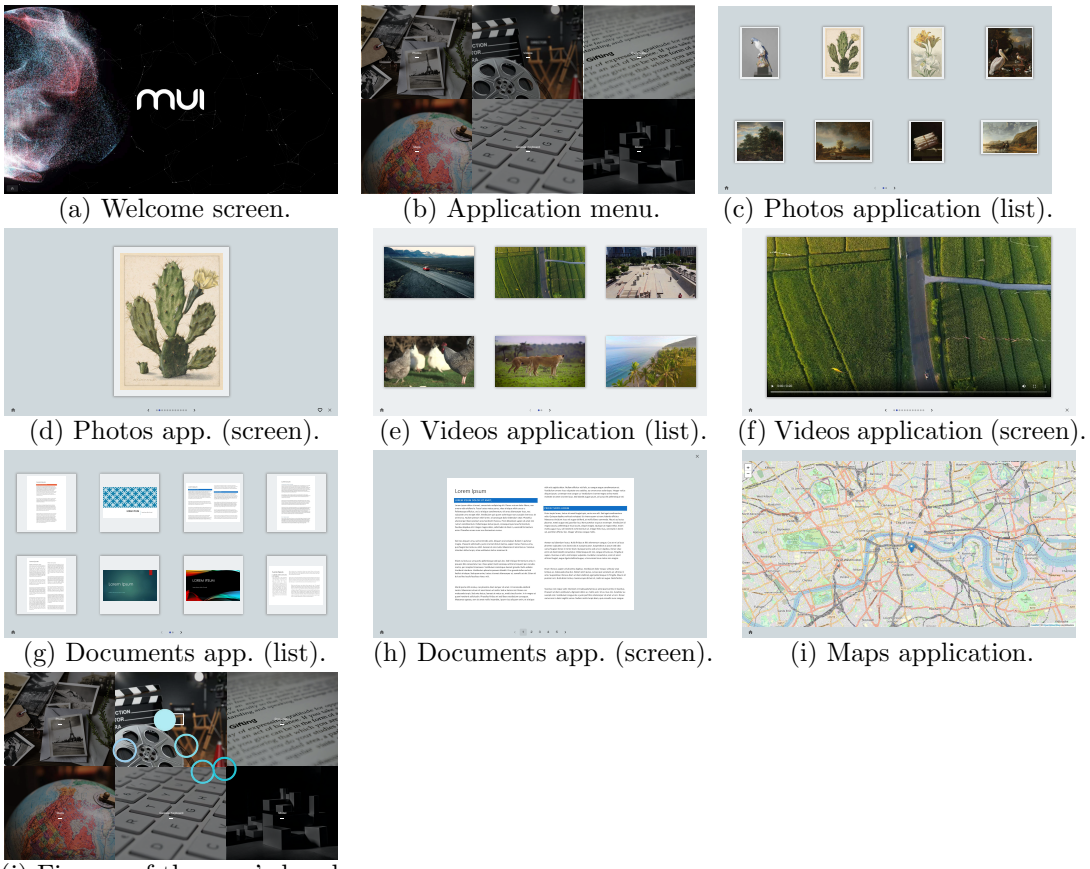
it is across different device types, but not across different media types, which all have their generic and specific sets of functions. While recent work reported that gestures elicited on different devices are not consistent, previous work did not investigate how to ensure consistency across media types.

3. LUI, an Application for Browsing Multimedia Objects on Large Displays

3.1. Overview of LUI

LUI is a gesture-based interactive application for manipulating multimedia objects on large displays, such as TVs and wall displays, in a semipublic environment. It currently focuses on interaction with photos, videos, documents, and maps, for which different sets of functions are desired (Table 1). LUI is designed to be widely distributed and easily installed by being cheap to set up and easy to use. As such, we settled on using the Leap Motion Controller (LMC)[1] for hand tracking, a cheap off-the-shelf sensor (less than 100$) that can be plugged via USB to the computer. We selected the LMC for mid-air gesture interaction because it tracks two hands and their ten fingers (Colgan, 2017) with high precision and robustness (Weichert et al., 2013), including the palm vector and hand radius. This sensor is comprised of two cameras and three infrared (IR) LEDs. The interaction space is .6m wide by .6m long by .6m deep, resulting in an .22 cubic meter volumetric space which takes the shape of an inverted pyramid above the sensor. The LMC API (Spiegelmock, 2013) enables to natively recognize a handful of system-defined gestures (Brandon, 2014), but custom gesture recognition algorithms often support a wider range of gestures (Bachman et al., 2018). The LMC benefits from a better accuracy and precision than other devices, such as a MS Kinect (Guzsvinecz et al., 2019) or an Intel RealSense (Khalaf et al., 2019). It is also widely used in many

[1]See `https://www.ultraleap.com/product/leap-motion-controller/`



(a) Welcome screen. (b) Application menu. (c) Photos application (list).

(d) Photos app. (screen). (e) Videos application (list). (f) Videos application (screen).

(g) Documents app. (list). (h) Documents app. (screen). (i) Maps application.

(j) Fingers of the user's hand.

Figure 2.: Screenshots of the third prototype of the Lui application.

domains of applications such as augmented reality (López et al., 2015) and activity recognition (Marin et al., 2016). Several LMC-based gesture recognition algorithms exist (Bachman et al., 2018). No study actually compares them on the same basis, apart from (Filho et al., 2018) with three classical machine learning algorithms, but without any recent recognizer such as DeepGru (Maghoumi & LaViola, 2019).

3.2. The LUI Components

Lui is developed with ReactJS, a popular open-source library for developing single-page applications (Boduch, 2019). ReactJS allows creating applications as a set of independent components. Lui currently consists of six components, including four applications, forming its functional core:

- Welcome screen (Fig. 2a): the entry point of Lui.
- App menu (Fig. 2b): a screen listing all Lui applications. It is designed to be extensible as new applications could be added in the future.
- Photos app (Fig. 2c): an application accessing galleries of pictures. Each photo can viewed in full-screen (Fig. 2d), where additional commands are available, such as zoom in/out, pan, rotate, like/dislike the picture, or go to the previous/next picture in the gallery and to the previous/next gallery of pictures.
- Videos app (Fig. 2e): an application accessing galleries of videos. Each video can be opened in full-screen mode (Fig. 2f), where additional commands are available, such as play/pause the video, change the volume, navigate in the video, or go to the previous/next video in the gallery and to the previous/next gallery of videos.

- Documents app (Fig. 2g): an application for manipulating textual or multimedia documents, such as reports, articles, and presentations. Each document can be viewed in full-screen (Fig. 2h), where commands are available, such as zoom in/out, pan, rotate, or go to page.
- Maps app (Fig. 2i): an application for accessing a map of the world. Actions such as zoom in/out and pan are available.

Lui is modular and extensible, as all applications are self-contained and autonomous so that they can be added or removed without impacting the other applications. To compensate the problems posed by fine-grained finger pointing, users' hands are depicted as enlarged circles for each finger on the display (Fig. 2j), as recommended (Koutsabasis & Domouzis, 2016). This provides feedback about the position of their hands and enables interaction with the elements displayed.

3.3. LUI Requirements

To address the quality of the gesture interaction, Lui should fulfill three requirements:

(1) *Consistency.* Similar gestures should be assigned consistently to similar functions, independently of the media type, to foster knowledge transfer (Dingler et al., 2018), to maximize gesture memorability and to reduce the learning curve (Delamare et al., 2019). Distinguishable gestures should be assigned to other specific functions (Kohlsdorf & Starner, 2013).

(2) *Continuity.* End users should be able to perform gestures continuously, without needing to notify the system when they are performing a gesture, *i.e.*, the system should continuously recognize gestures and distinguish intended gestures from parasitic motion with little to no user intervention. End users should be able to manipulate media objects directly with a reasonable system response time (*e.g.*, 1 sec for direct manipulation according to (Nielsen, 1994)) and a continuous feedback to the user (Aigner et al., 2012) (*e.g.*, gradually zooming in a picture by increasing the distance between the index and the thumb). Whether a gesture is discrete for a 0D/1D function or continuous for a 2D/3D function, this should not induce any difference for the end user.

(3) *Customizability.* The system should support users in including their own gestures without requiring any modification to the Lui core (Coyette et al., 2007). User-defined gesture sets (Grijincu et al., 2014) usually result in higher satisfaction and memorability than system- or designer-defined gestures (Nacenta et al., 2013). Some gesture recognizers perform better when the gestures used for training are recorded by the target user of the system (Vatavu, 2013).

3.4. LUI Iterative Design

Designing a gesture set satisfying the three requirements was a major objective in the development of Lui. This section summarizes the three iterations of Lui and its gesture set (Fig. 3).

3.4.1. First Iteration (v0)

In this first prototype (Parthiban & Lee, 2019), the basics of Lui were laid out. Users could navigate through lists of photos and videos, switch to full-screen mode, play/-

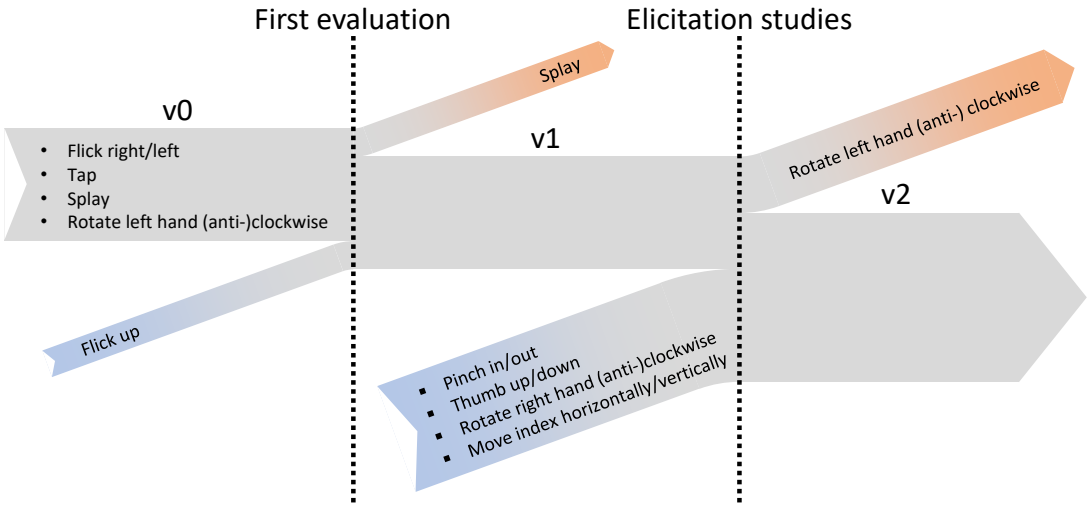


Figure 3.: Evolution of the gesture set during the iterative design.

pause a video, or change the volume of a video. Gesture recognition was developed opportunistically based on the LMC by design patterns (Spiegelmock, 2013). Adding new functions or gesture classes was therefore more complex since the gesture recognition was programmed manually and prevented end-users from changing the gestures assigned to specific functions. As such, the first version of the gesture set was rather straightforward, but limited.

3.4.2. Second Iteration (v1)

A usability testing was conducted with 20 participants to evaluate their subjective satisfaction regarding the first Lui prototype (Parthiban & Lee, 2019; Dumas et al., 2019). It highlighted multiple issues, including a high learning curve, unpredictable gesture recognition, and fatigue, in particular for the Splay gesture, which participants found particularly difficult to perform and to reproduce accurately. Observed usability issues were fixed, such as: the end-user manipulates contents with four fingers together instead of the index finger alone, the swipe up gesture works best for small motions, the opening/pausing of videos should be very sensitive. Some participants expressed their intention to define their own preferred gestures, instead of using the predefined ones. In this second prototype, the Splay gesture was replaced by the Flick up gesture, which is easier to perform. The other issues highlighted by the testing were left unaddressed in this version, as they required more substantial modifications to the Lui core.

3.4.3. Third Iteration (v2)

Based on the assets of the first two prototypes, a third prototype was developed, featuring a new modular pipeline for gesture recognition (see Section 4). This pipeline enables end-users to add their own preferred gestures to predefined ones without having to modify the Lui core. It also simplifies the process of extending Lui with new applications and gestures. To demonstrate this, some functions were added to the Photos and Videos applications (*e.g.*, rotating a photo and browsing through a video), and new Documents and Maps applications were developed.

4. LUI Development Method

We followed a three-stage process for the development of the third iteration of LUI. First, a contextual GES was conducted to design a consistent and highly guessable gesture set. Then, a pipeline was designed for gesture recognition. Finally, a comparative testing of recognizers served to determine the best pipeline configuration for this gesture set.

4.1. LUI Gesture Set

4.1.1. Contextual Gesture Elicitation Study

As the number of functions increased, it was decided to conduct a GES (Wobbrock et al., 2009) to explore a consensus set of consistent gestures that would optimize agreement among end-users. Based on the classical method (Wobbrock et al., 2009) and some extensions (Gheran et al., 2018), we initially conducted two GES (see Appendix C): one for photo browsing and another for video browsing. Prior work (Anthony et al., 2013; Gronewald et al., 2016) pointed out that gesture sets for different devices could be inconsistent. Our two studies confirm this observation as they led to inconsistent gestures tailored to one media type at a time. Rather than conducting a separate GES for each media type, we conducted a single GES (see Appendix A) integrating the various media with both generic and specific functions (see Table 1). Typical GES protocols present referents to participants as a textual prompt to minimise any bias or with visual priming (Morris, 2012; Morris et al., 2014) to provide them with some guidance about the state before and after issuing the gesture proposal. Unlike these approaches, we introduced a *contextual gesture elicitation study*: participants directly interact with a custom software which exactly executes the referents' functions (see Table 1) to establish contextual priming beyond mere visual priming, since priming improves learnability and memorability (Ali et al., 2021). For example, to elicit a gesture for playing a video from a participant, the software includes a video control whose functions can be directly mapped to the real gestures proposed (Fig. A1).

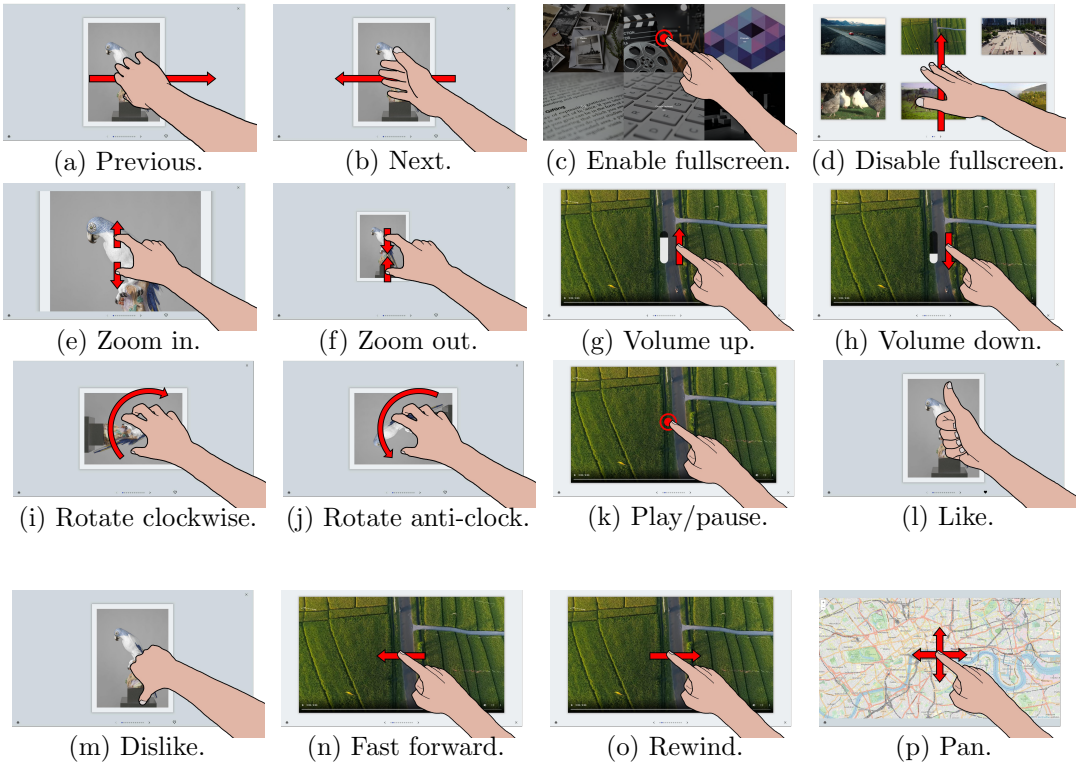
The result of our contextual GES consists of a characterization of users' gesture input behaviors with valuable information for designers, practitioners, and end-users regarding the consensus among participants (which is computed as an *agreement* and *co-agreement rate* (Vatavu & Wobbrock, 2015)), most frequent (thus, generalizable across users) gestures for a specified task, and insights into users' conceptual models for performing tasks. (Dingler et al., 2018) computed the agreement score (Wobbrock et al., 2009) for their multi-device GES. Instead, we computed the agreement rate $AR(r)$ (Vatavu & Wobbrock, 2015), which is smaller and more restrictive than the agreement score $A(r)$, as follows:

$$A(r) = \sum_{P_i \subseteq P} \left(\frac{|P_i|}{|P|} \right)^2 \geq AR(r) = \frac{\sum_{P_i \subseteq P} \frac{1}{2}|P_i|\,(|P_i|-1)}{\frac{1}{2}|P_i|\,(|P_i|-1)} \tag{1}$$

where r denotes the referent to elicit a gesture, $|P|$ denotes the number of elicited gestures, and $|P_i|$ denotes the number of gestures elicited for the i^{th} subgroup of P.

Table A1 summarizes the two most agreed gestures for each referent for the contextual GES for all media. From these results, we observe that the majority ($\frac{16}{18}$=88.9%) of the most agreed gestures were one-handed. One-handed gestures let users interact with the system while holding an object in the other hand (*e.g.*, a glass of water or a paper

document). In addition, the low occurrence of two-handed gestures in our study may be explained by a lower ease of execution compared to one-handed gestures (Morris et al., 2010). Some referents, such as enable/disable subtitles, have a very low agreement rate. For these referents, user-elicited gestures may not be the most appropriate, as the low consensus may result in lower guessability and memorability (Wobbrock et al., 2005). Instead, it may be preferable to favor other solutions, such as prompting users to assign their own gesture to these actions at the start of the application (*i.e.*, user-defined gestures (Nacenta et al., 2013)), and/or providing an indirect way of triggering these actions (*e.g.*, by navigating through a menu with highly agreed gestures). For referents with a low to medium agreement rate, such as play and pause, aliasing (*i.e.*, assign multiple gestures, such as the most and second most agreed gestures, to one referent) may greatly increase guessability (Wobbrock et al., 2005, 2009). This would enable users to use the gesture they are most familiar with, and thus flatten the learning curve and increase memorability of the gesture set. However, attentive care should be given to avoid to avoid overloading gestures.



(a) Previous. (b) Next. (c) Enable fullscreen. (d) Disable fullscreen.

(e) Zoom in. (f) Zoom out. (g) Volume up. (h) Volume down.

(i) Rotate clockwise. (j) Rotate anti-clock. (k) Play/pause. (l) Like.

(m) Dislike. (n) Fast forward. (o) Rewind. (p) Pan.

Figure 4.: Illustrations of gestures mapped on each function in the Lui application.

4.1.2. Gesture Set Design

The Lui gesture set (Fig. 4 and Table 2) resulting from the contextual GES consists of gestures selected following four guidelines inspired by the literature (Morris et al., 2010; Wobbrock et al., 2005, 2009; Dingler et al., 2018):

(1) Favor the most agreed gestures, as this usually results in flatter learning curve and higher memorability (Wobbrock et al., 2005, 2009; Dingler et al., 2018). For instance, we selected "flick left" instead of "flick right" for the "next" function,

as the former was proposed by more participants.

(2) Prefer gestures that are less physically and mentally demanding to limit frustration and fatigue (Morris et al., 2010). For example, we favored one-handed over two-handed gestures.

(3) Map similar gestures to similar functions, independently of the media to increase memorability and flatten the learning curve (Dingler et al., 2018). For instance, we selected the same gesture for zooming in a photo/in a map and used symmetrical gestures for going to the previous/next item in a list.

(4) Map unique gestures to different functions occurring in a same context to avoid conflicts where one gesture would trigger multiple functions (Wobbrock et al., 2005; Dingler et al., 2018).

Together, these guidelines enabled us to build a conflict-free, highly guessable, and memorable gesture set which minimizes end-users' mental and physical effort. The "pan" gesture was added later as part of the Maps application, but was not part of the elicitation study.

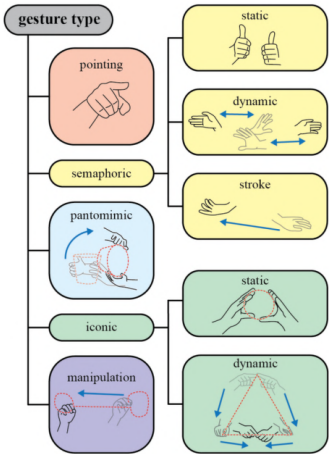


Figure 5.: Aigner classification.

	Elicitation		Recognition	
Action	Gesture	Category	Name	Type
Previous	Flick right	Stroke semaphoric	Flick right	Traj./dynamic
Next	Flick left	Stroke semaphoric	Flick left	Traj./dynamic
Enable fullscreen	Tap	Pointing	Tap	Traj./dynamic
Disable fullscreen	Flick up	Stroke semaphoric	Flick up	Traj./dynamic
Zoom in	Pinch out	Manipulation	Pinch	Pose/static
Zoom out	Pinch in	Manipulation	Pinch	Pose/static
Volume up	Swipe up	Manipulation	Point index	Pose/static
Volume down	Swipe down	Manipulation	Point index	Pose/static
Rotate clockwise	Rotate a knob clockwise	Manipulation	Grab	Pose/static
Rotate anti-clockwise	Rotate a knob anti-clockwise	Manipulation	Grab	Pose/static
Play	Tap	Stroke semaphoric	Tap	Traj./dynamic
Pause	Tap	Stroke semaphoric	Tap	Traj./dynamic
Like	Thumbs up	Static semaphoric	Thumb	Pose/static
Dislike	Thumbs down	Static semaphoric	Thumb	Pose/static
Fast-forward	Swipe left	Manipulation	Point index	Pose/static
Rewind	Swipe right	Manipulation	Point index	Pose/static
Pan	–	–	Point index	Pose/static

Table 2.: Classification of Lui gestures.

4.1.3. Gesture Classification

After designing a consistent and memorable gesture set, the next step was to implement an appropriate pipeline for recognizing these gestures. As such, we first attempted to classify each gesture according to Aigner *et al.*'s HCI-oriented gesture taxonomy (Aigner et al., 2012). While several multidisciplinary taxonomies exist in general for classifying gestures, including Kendon's taxonomy (Kendon, 1988) and McNeil's taxonomy (McNeil, 1995), we rely on Aigner's classification as it identifies gesture classes suitable to the specific field of HCI by empirically validating 5,500 collected gestures on interactive functions without accompanying them with speech. Indeed, (Ekman, 1969) characterize human gestures according to three fundamental considerations, *i.e.*, origin, usage, and coding, and structure them into idiosyncratic, interactive, informative, and communicative classes. Kendon (Kendon, 1988) structures gestures along a continuum based on speech: gesticulation (beat, cohesives), language-like (iconic), pantomimes, emblems (deictic), and sign language (symbolic). McNeill (McNeil, 1995) structures gestures into iconics, metaphorics, beats, cohesives, and deictics. By analyzing each

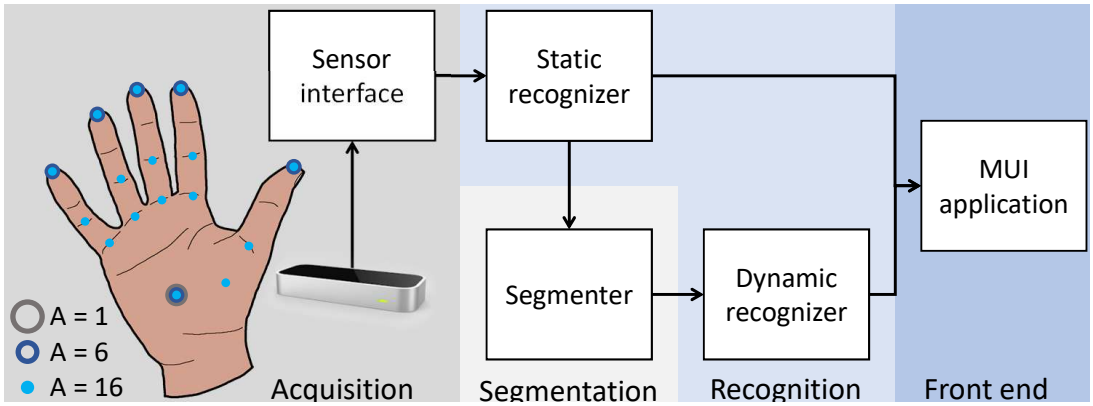


Figure 6.: The Lui gesture recognition pipeline.

gesture and its effect on the application, this step helped us create a gesture recognition pipeline that is well-suited to the Lui application. Table 2 summarizes the results of our classification and details how each gesture should be recognized.

Eight of the 16 elicited gestures are classified in the *manipulation* category, which are gestures for manipulating media objects in a short feedback loop. The gesture effect on the object is perceived in real time while the user is performing it, with low latency (Aigner et al., 2012) (*e.g.*, a response time of 0.1 sec to perceive interaction as a direct manipulation and 1 sec to keep it seamless (Nielsen, 1994)). Other authors estimate the response time as follows: (Cheng, 2017) situates the best compromise around 500 ms with a recognition rate above 90%, while (Annett et al., 2014) report that end-users perceive a response time of 50 ms for inking tasks on a touch-sensitive surface. As an example of direct manipulation, the size of an image should change proportionally to the thumb-index distance while the user is performing a pinch in or pinch out gesture. For these gestures, the hand pose conveys the meaning of the gesture. They can thus be recognized by analysing the user's hand pose at each frame in real time. Six gestures are considered as *stroke semaphoric* and two are considered as *static semaphoric*. *Semaphoric gestures* are hand poses and movements used to convey a specific meaning, often unrelated to the gesture. They can be further divided into three categories, including *static* (if the meaning is entirely conveyed by the hand pose) and *strokes* (which consist of single stroke hand motions) (Aigner et al., 2012). The *stroke semaphoric* gestures could be recognized by analyzing the hand trajectory over time, while the *static semaphoric* gestures could be recognized by analyzing the user's hand pose. One gesture belongs to the *pointing* category, *i.e.*, gestures which indicate objects and directions (Aigner et al., 2012). Indeed, to enable fullscreen, the tap gesture acts as a selection method, as users first need to hover their finger above the selected item before performing the gesture. This gesture could be recognized similarly to the *stroke semaphoric* gestures.

4.2. LUI Software Architecture and Implementation

The Lui software architecture relies on a flexible pipeline workflow for gesture recognition divided into the following steps, each being encapsulated into one or many alternative modules (Fig. 6):

(1) *Acquisition.* An LMC acquires raw data at the rate of 60 frames per second. The Sensor interface parses each frame into a simple, reusable format, and

Recognizer		Gesture class						Global accuracy		Time
Name	A	Flat	Point	Pinch	Grab	Thumb	Idle	Mean	SD	
$\$P^3+$	6	99.7%	100.0%	100.0%	100.0%	100.0%	100.0%	99.9%	0.1%	0.15ms
$\$P^3+$	16	100.0%	99.9%	99.7%	100.0%	100.0%	100.0%	99.9%	0.1%	0.46ms
GPSDa	6	99.6%	99.0%	98.3%	99.9%	98.2%	100.0%	99.2%	0.8%	0.04ms
GPSDa	16	99.6%	99.8%	99.4%	100.0%	99.5%	100.0%	99.7%	0.3%	0.22ms

Table 3.: Recognition rate of each static recognizer, for T=128. Accuracy higher than 98% is highlighted in blue.

forwards them to the next module.

(2) *Recognition.* The STATIC RECOGNIZER analyses each frame to extract the hand pose (*i.e.*, static gesture) contained, whereas the DYNAMIC RECOGNIZER analyses a sequence of frames to determine the hand motion (*i.e.*, dynamic gesture) contained in that sequence. Disambiguating static from dynamic gestures represents a key factor to send the gesture candidate to the right recognizer(s) by selecting algorithms specifically tailored for each gesture type. Each training dataset can be modified without affecting the rest of the pipeline. End-users can change the gesture assigned to a function and developers have support for new gestures.

(3) *Segmentation.* Before a dynamic gesture can be recognized, it first needs to be extracted from the raw stream of data. The SEGMENTER collects frames from the LMC and extracts (*i.e.*, segments) the sequences of frames that correspond to a gesture intended by the user.

(4) *Front end.* Gestures can be assigned effects on the UI that are triggered each time the gesture is recognized. For instance, the Pinch static gesture rotates a picture depending on the rotation of the hand and the tap dynamic gesture open an application, display fullscreen, or play/pause a video, depending on the context.

4.3. LUI Pipeline Configuration

We now motivate our choice of modules for the LUI gesture recognition pipeline. We conduct a comparative testing of static and dynamic gesture recognizers (see (Magrofuoco et al., 2021) for a survey in 2D) and discuss the use of a sliding window for gesture segmentation.

4.3.1. Static Gestures

Two static gesture recognizers were tested on a custom gesture set containing six types of poses (grab, pinch, point index, flat hand, thumb, and rest) acquired from one participant. Each pose was performed in various orientations and scales, resulting in more than 1000 templates per class. There are three independent variables:

- RECOGNIZER: a nominal variable with two conditions, representing the static recognizers tested (Appendix B.1): GPSDa, a custom recognizer with α=0.65) and $\$P^3+$, adapted from (Vatavu, 2017).
- NUMBER OF ARTICULATIONS: a numerical variable with two conditions, representing the number of articulations for the recognizers (Fig. 6): $A \in \{6, 16\}$.
- NUMBER OF TEMPLATES: a numerical variable with eight conditions, representing the number of templates used to train the recognizers: $T \in \{1, 2, 4, 8,$

16, 32, 64, 128}.

The testing procedure is as follows: for each recognizer, for each value of A and T, and for each gesture class, we repeat 1000 times: (1) select a random testing template from the gesture set; (2) train the recognizer using T randomly selected templates, produced by the same user as the testing template (user-dependent testing); and (3) recognize the testing template. This experiment shows that GPSDa behaves differently than $\$P^3+$ (Table 3): GPSDa performs better with a larger number of articulations, while the opposite occurs for $\$P^3+$. GPSDa performs better than $\$P^3+$ with a small number of templates (*e.g.*, with 4 templates, GPSDa (A=16) reaches 91.0% accuracy against 87.1% for $\$P^3+$ (A=6)), but $\$P^3+$ starts performing better as the number of templates increases. Both recognizers achieved >99% accuracy with 128 training templates. Despite $\$P^3+$ being very slightly better in the benchmark, GPSDa was selected for the Lui application as its rotation invariance gives it an edge in real-world situations (*e.g.*, when the user's hand is rotated relative to the training samples).

4.3.2. Dynamic Gestures

Similarly to the static gesture recognizers, eight dynamic gesture recognizers were tested on the Lui gesture set to determine the best suited recognizer. The gesture set contains four types of pre-segmented dynamic gestures, *i.e.*, flick left, flick right, flick up, and tap, acquired from four participants. Gestures were performed four times by each participant, giving 16 templates per class. There are three independent variables:

- Recognizer: a nominal variable with five conditions, representing the dynamic recognizers tested: 3¢ (Caputo et al., 2017), $\$P^3$ (adapted from (Vatavu, 2012)), $\$P^3+$ (adapted from (Vatavu, 2017)), $\$Q^3$ (adapted from (Vatavu et al., 2018)), and Jackknife (Taranta et al., 2017). The recognizers are detailed in Appendix B.2.
- Number of Articulations: a numerical variable with three conditions, representing the number of articulations taken into account by the recognizers (Fig. 6): $A \in \{1, 6, 16\}$.
- Number of Templates: a numerical variable with four conditions, representing the number of templates used to train the recognizers: $T \in \{1, 2, 4, 8\}$.

The testing procedure is similar to the one used for static recognizers: for each recognizer, for each value of A and T, and for each gesture class, we repeat the three following steps 1000 times: (1) select a random testing template from the gesture set; (2) train the recognizer using T randomly selected templates produced by different users than the testing template (user-independent testing); and (3) attempt to recognize the testing template. Each template was first resampled to 16 points.

Because the dataset includes only four simple gestures, the recognition rates are very high for most recognizers even with very few training templates (Table 4, T=8). Jackknife already reaches 100% with one training template, and accuracy for all recognizers generally increases with the number of training templates. Jackknife (A=1) was selected as the dynamic recognizer for the Lui application, as it maximizes accuracy and minimizes the execution time on this dataset.

4.3.3. Gesture Segmentation

A simple sliding window was selected for gesture segmentation. To limit the number of false positives, it was combined with two other techniques: (1) a motion rejection threshold to reject small hand motions such as hand tremors, and (2) a rejection

Recognizer		Gesture class				Global accuracy		Time
Name	A	Flick up	Flick left	Flick right	Tap	Mean	SD	
3¢	1	100.0%	97.2%	100.0%	100.0%	99.3%	1.4%	0.78ms
$P^3	1	100.0%	67.7%	70.1%	100.0%	84.5%	18.0%	0.33ms
$P^3	6	94.3%	91.5%	80.3%	99.9%	91.5%	8.2%	0.38ms
$P^3	16	91.9%	99.9%	90.8%	100.0%	95.7%	5.0%	0.28ms
$P^3+	1	100.0%	58.9%	53.5%	100.0%	78.1%	25.4%	0.05ms
$P^3+	6	99.9%	85.8%	78.1%	100.0%	91.0%	10.9%	0.06ms
$P^3+	16	100.0%	94.2%	83.0%	100.0%	94.3%	8.0%	0.10ms
$Q^3	1	100.0%	65.9%	36.1%	100.0%	75.5%	30.8%	0.91ms
$Q^3	6	100.0%	82.1%	71.6%	100.0%	88.4%	14.0%	1.03ms
$Q^3	16	99.2%	90.8%	75.2%	100.0%	91.3%	11.5%	0.97ms
Jackknife	1	100.0%	100.0%	100.0%	100.0%	100.0%	0.0%	0.04ms
Jackknife	6	100.0%	100.0%	100.0%	100.0%	100.0%	0.0%	0.10ms
Jackknife	16	100.0%	100.0%	100.0%	100.0%	100.0%	0.0%	0.19ms

Table 4.: Recognition rate of each dynamic recognizer, for T=8. Recognizers with a global accuracy higher than 98% are highlighted in blue.

threshold based on the score returned by the recognizers, to reject gestures that are performed poorly. Sliding window is easy to implement, fast to execute, requires no specific user input, and has already been combined with recognizers such as Jackknife (Taranta et al., 2017) and 3¢ (Caputo et al., 2019). However, it is very sensitive to parasitic gestures, which may hinder the user experience. In the future, more advanced segmentation techniques (Kratz & Back, 2015; Chen et al., 2016; Taranta et al., 2021) could be incorporated into a new module without requiring any code modification.

5. LUI Final Evaluation

This section reports the results of an experiment performed on the third Lui version, according to a procedure inspired by (Vogiatzidakis & Koutsabasis, 2022).

5.1. Participants

A stratified sampling of 17 participants (8 females and 9 males), aged between 22 and 65 years old (M=37.5, SD=15.3 years) was randomly selected from a list of volunteers to balance between genders, backgrounds, and ages (Table 5). All participants were right-handed and all of them used smartphones on a daily basis. All but one participant ($\frac{16}{17}$=94.1%) used computers at least once a week, and only two participants ($\frac{2}{17}$=11.8%) frequently used gestural interaction. Occupations included student, researcher, employee, computer scientist, physiotherapist, and accountants.

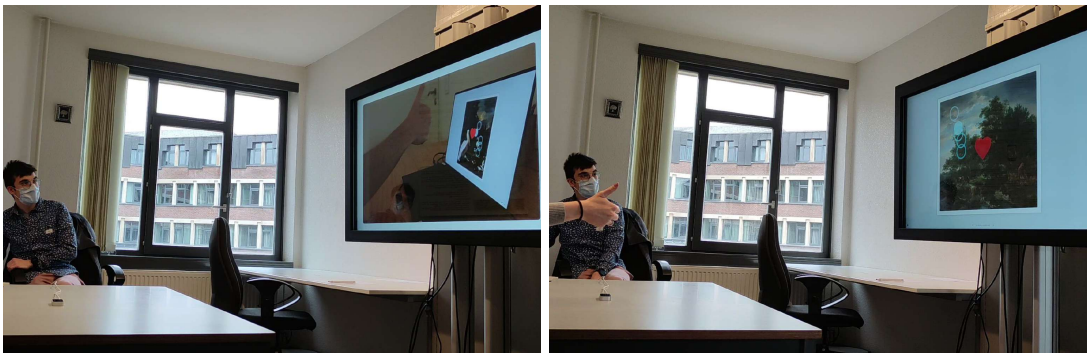
5.2. Apparatus

The experiment involved multiple devices, including a large TV screen for displaying the Lui application and tutorial videos (Fig. 7), an LMC for capturing the participants' gestures, and a laptop for running the Lui application and controlling the experiment

Gender	Female				Male			
Background	Technical		Non-technical		Technical		Non-technical	
Age	≤ 30 y.o.	> 30 y.o.	≤ 30 y.o.	> 30 y.o.	≤ 30 y.o.	> 30 y.o.	≤ 30 y.o.	> 30 y.o.
#participants	2	2	2	2	2	2	2	3

Table 5.: Distribution of the participants in our study.

with a custom software. A camera was positioned to record both the participants' hands and the TV screen. The Lui application was configured to record the logs of the gestures recognized by the system. Participants were installed at a desk and faced a large screen situated at around two meters on the opposite side of the desk. The LMC was positioned in front of them.



(a) Tutorial video playing on the screen. (b) Participant performing a gesture.

Figure 7.: The setup of the experiment.

5.3. Protocol

The experiment consisted of two sessions conducted on separate days. The interval between the first and the second sessions of a participant ranged from 3 to 14 days (M=7, SD=2.2 days).

5.3.1. Session 1 (Discovery and Short-term Memorability)

The first session was aimed at introducing Lui to participants and evaluating its user experience and the short-term memorability of its gesture set. It consisted of six phases:

(1) *Consent form and demographic information.* The participants were introduced to the procedure and informed that they could choose to leave the experiment at any time. They were then invited to sign a GDPR-compliant consent form and fill in a demographic survey with information such as their age, gender, level of education, occupation, digital skills (Rose et al., 2012), and previous experience with technologies such as smartphones, computers, and gestural interaction.
(2) *Discovery phase.* Participants interacted freely with the Lui application for three minutes, starting from the app menu. They did not receive any specific instructions or information about the interface, except that it was controllable using the movements of the right hand above the sensor, without touching any surface. This phase served as a first introduction of Lui to the participants.
(3) *Learning phase.* This phase served to teach the gestures to the participants and let them familiarize with the system. The participants were given 17 tasks in

Id	Task	Target(s)	Gestures
A1	Previous	Photo/video/document/page	Flick right
A2	Next	Photo/video/document/page	Flick left
A3	Enable fullscreen	Photo/video/document/app	Tap
A4	Disable fullscreen	Photo/video/document/app	Flick up
A5	Zoom in	Photo/document/map	Pinch out
A6	Zoom out	Photo/document/map	Pinch in
A7	Volume up	Video	Swipe up
A8	Volume down	Video	Swipe down
A9	Rotate clockwise	Photo/document	Rotate a knob clockwise
A10	Rotate anti-clockwise	Photo/document	Rotate a knob anti-clockwise
A11	Play	Video	Tap
A12	Pause	Video	Tap
A13	Like	Photo	Thumbs up
A14	Dislike	Photo	Thumbs down
A15	Fast-forward	Video	Swipe left
A16	Rewind	Video	Swipe right
A17	Pan	Map	Move hand while pointing index

Table 6.: The 17 atomic tasks, potential target content, and associated gestures.

a random order (Table 6). Each task was read aloud by the experimenter, who then played a short tutorial video of the associated gesture (Fig. 7a). The use of videos ensured that all participants received the same instructions. Once the tutorial finished playing, the experimenter loaded the associated page of the Lui application and prompted the participants to perform the gesture in order to complete the task (Fig. 7b). Participants could re-watch the tutorial videos as many times as necessary. If a participant accidentally left the current page (*e.g.*, by performing the wrong gesture), the experimenter informed them and re-loaded the correct page.

(4) *Resting phase.* Participants were given five minutes to rest in a separate room.

(5) *Testing phase.* This phase aimed to evaluate the short-term memorability of the gesture set by asking participants to perform 20 tasks with no external help. The participants first completed 17 atomic tasks (Table 6) in a random order, following a similar procedure to the learning phase: the experimenter first read the task aloud, loaded the associated page of the Lui application, and prompted the participants to perform the gesture. The target content was randomly selected for each task (*e.g.*, for the "zoom in" task, the target could be a picture, a document, or a map) and the tasks were grouped by target content. The participants then performed three compound tasks (Table 7). For each compound task, the experimenter read the task aloud, loaded the application menu, and prompted the participants to perform the task. Participants could request the experimenter to repeat the instructions or to go back to the application menu.

(6) *Questionnaire.* After completing the testing phase, participants were asked to complete a questionnaire consisting of seven UEQ+ scales (Section 5.4).

5.3.2. Session 2 (Long-term Memorability)

Session 2 was much shorter than the first session and consisted of only three phases:

Id	Task	Instructions	Required actions
C1	Maps	In the Maps app, move and enlarge the map so that the two blue markers are situated at the left and right sides of the screen.	Enable fullscreen, zoom in, pan
C2	Photos	In the Photos app, manipulate the windmill painting so that the author's name is the right way up and large enough to be easily readable.	Enable fullscreen, next, zoom in, rotate clockwise and/or anticlockwise
C3	Videos	In the Videos app, find the words pronounced halfway through the "sea turtles" video. Fast-forward through the video to get to the right part.	Enable fullscreen, next, play, fast-forward

Table 7.: The three compound tasks and the minimum set of actions required to complete each task.

(1) *Introduction.* Participants were welcomed and reminded of the procedure, in particular, that they could leave the experiment at any time.
(2) *Testing phase.* This second testing phase served to evaluate the long-term memorability of the Lui gesture set. We followed the same procedure as in the testing phase of the first session.
(3) *Questionnaire.* At the end of the session, participants were asked to fill in the same UEQ+ questionnaire (Schrepp et al., 2019) as they did in the first session.

5.4. Metrics and Data Collection

We collected a set of metrics for the learning and testing phases of both sessions to give us insight into the performance and usability of the system:

(1) The Task Success rate measures the percentage between the number of successful tasks (*i.e.*, the gesture is correctly remembered, produced, and recognized) and the total number of executed tasks. An atomic task was considered successful if it was correctly executed within a maximum of four trials. Regarding compound tasks, the number of trials was not limited since they required participants to perform several gestures in an undefined order. We thus considered a compound task unsuccessful only if it was skipped by the participant.
(2) The Number of trials measures the total number of gestures executed for each task, whether successful, incorrectly recalled, incorrectly executed or incorrectly recognized by the system.

In addition to these metrics, we collected demographic information from the participants at the start of the first session. Participants were instructed to filled in at the end of each session a UEQ+ questionnaire (User Experience Questionnaire) (Schrepp et al., 2017), a modular extension of UEQ in which we selected 7 scales (*i.e.*, attractiveness, efficiency, perspicuity, dependability, stimulation, usefulness, and intuitive use) among 20 scales to focus on evaluating the user experience of participants interacting by gestures with Lui. Each scale is in turn decomposed into four subscales to be evaluated (*e.g.*, attractiveness is decomposed into four subscales: annoying *vs.* enjoyable, bad *vs.* good, unpleasant *vs.* pleasant, and unfriendly *vs.* friendly), each subscale being a differential scale with 7 points between items of each pair (*e.g.*, annoying o o o o o o o enjoyable). Finally, we captured video footage of the discovery, learning, and testing phases of the experiment.

5.5. Results and Discussion

5.5.1. Task Success Rate

Table 8 gives the success rate for the atomic and compound tasks. Only two atomic tasks have a success rate below 80% in the learning phase: tasks A4 (disable full screen) and A9 (rotate clockwise). In the first testing phase, only task A9 has a success rate below 80%, while in the second testing phase, the overall success rate decreases slightly, and three tasks have a success rate lower than 80%: tasks A9, A10 (rotate anticlockwise), and A16 (rewind). Overall, the lower success rate of the rotating (anti-)clockwise tasks may be explained by the high sensitivity of the system to the users' hand pose. For instance, if the users' fingers were too far apart when performing the "grab" gesture, it was not recognized by the system. To a lesser extent, this high sensitivity may have affected the recognition of other static gestures (see Table 2), such as in the "fast-forward", "rewind", and "pan" tasks.

Regarding compound tasks, tasks C1 (photo) and C2 (video) had a higher success rate than task C3 (map) in both the first and second testing phases. The "pan" gesture used in the Maps application was not elicited, which may have impacted the usability of the application. In addition, this task required more precision than the other two, as it required participants to accurately manipulate the map so that two markers were situated on each side of the screen.

Id	**Task**	**Learning phase**	**Testing phase 1**	**Testing phase 2**
A1	Previous	100%	100%	100%
A2	Next	100%	94%	94%
A3	Enable fullscreen	88%	82%	100%
A4	Disable fullscreen	76%	88%	100%
A5	Zoom in	100%	88%	94%
A6	Zoom out	94%	94%	82%
A7	Volume up	94%	88%	82%
A8	Volume down	82%	100%	94%
A9	Rotate clockwise	71%	71%	71%
A10	Rotate anti-clockwise	82%	82%	71%
A11	Play	82%	88%	100%
A12	Pause	94%	88%	82%
A13	Like	100%	100%	100%
A14	Dislike	100%	100%	94%
A15	Fast-forward	88%	88%	88%
A16	Rewind	82%	82%	71%
A17	Pan	88%	88%	88%
C1	Photo	–	100%	94%
C2	Video	–	82%	94%
C3	Map	–	76%	65%

Table 8.: Success rate for the 17 atomic and 3 compound tasks.

5.5.2. Number of Trials

Table 9 gives the average and median number of tries needed to successfully complete the atomic tasks. These metrics are presented for the learning session and the two

Id	Task	Learning phase		Testing phase 1		Testing phase 2	
		Average	Median	Average	Median	Average	Median
A1	Previous	1,2	1	1,6	1	1,7	1
A2	Next	1,1	1	1,3	1	1,6	2
A3	Enable fullscreen	1,5	1	1,5	1	1,1	1
A4	Disable fullscreen	1,2	1	1,7	2	1,5	1
A5	Zoom in	1,4	1	1,3	1	1,5	1
A6	Zoom out	1,4	1	1,1	1	1,2	1
A7	Volume up	1,3	1	1,3	1	1,1	1
A8	Volume down	2,0	2	1,5	1	1,3	1
A9	Rotate clockwise	2,1	1,5	1,6	1,5	1,9	1,5
A10	Rotate anti-clockwise	1,8	1,5	2,1	1,5	2,4	2,5
A11	Play	1,3	1	2,0	1	1,6	1
A12	Pause	1,4	1	1,5	1	1,8	1,5
A13	Like	1,0	1	1,2	1	1,3	1
A14	Dislike	1,0	1	1,1	1	1,0	1
A15	Fast-forward	1,7	1	1,7	1	1,9	2
A16	Rewind	1,4	1	1,7	1	1,4	1
A17	Pan	1,3	1	1,3	1	1,5	1

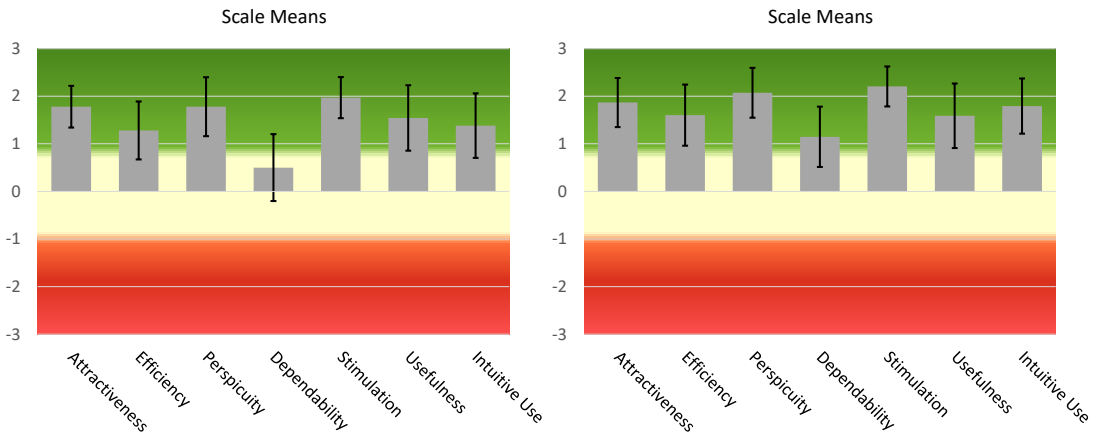
Table 9.: Number of trials for successful completion of each atomic task.

testing sessions, allowing in particular to study the memorability of gestures.

As presented, most tasks were completed with a single try. The most problematic tasks were the rotations, whether clockwise or anticlockwise. We can also observe that more attempts were needed to pass the tasks in session 2, especially for A1 (previous), A2 (next), A12 (pause) and A15 (fast forward). On the contrary, the tasks of increasing or decreasing the volume (A7 and A8) required fewer tries over time.

5.5.3. Results of the UEQ+ Questionnaire

Users' answers to the 7 selected UEQ+ scales were computed and interpreted with the UEQ data analysis tool. Fig. 8 represents the mean score for each selected scale on an interval [-3,...,+3], where a value of -3 expresses the weakest value and +3 expresses the strongest value for both sessions (first in Fig. 8a and second in Fig. 8b). If we stick to the interpretation that a value greater than or equal to 0.8 represents a positive evaluation (Schrepp et al., 2017), then all mean scores for session 1 are considered positive (ranging from M=1.28 for efficiency to M=1.97 for stimulation), except dependability (M=0.50, SD=2.22) considered as a neutral value. This low value can be explained by the cognitive destabilization of participants producing these kinds of gestures for the first time, and being puzzled by them. However, this phenomenon is observed transiently insofar as the gestures, initially felt to be difficult to produce because they are unknown, then become more easy to produce. The dependability experiences a positive increase of 130% (M=1.15, SD=1.80) in session 2. Mean scores for session 2 are all superior to those obtained for session 1 and range from M=1.15 for dependability to M=2.21 for stimulation. The mean score for some scales even largely increased, such as efficiency (from M=1.28 to M=1.60), perspicuity (from M=1.78 to M=2.07), and intuitive use (from M=1.38 to M=1.79). The KPI (Key Performance Indicator) computed by UEQ+ Hinderks et al. (2019) started from 1.44 for session 1 to 1.74 for session 2, which confirms the overall positive evolution.



(a) Session 1. (b) Session 2.

Figure 8.: Mean score for each scale of the UEQ+ questionnaire.

6. Discussion of LUI Benefits and Limitations

6.1. Benefits

LUI satisfies the three requirements defined in Section 3.3. Each requirement is now revisited in the light of this work.

6.1.1. Consistency

The LUI gesture set was designed to be consistent across the various types of media contents, by assigning similar gestures to similar functions, regardless of the media type. In addition, it is expected to be compliant with end-users' expectations as the gesture set was based on a contextual gesture elicitation study. For this purpose, we defined new media-dependent agreement rates which result into calculating a (in-)consistency rate.

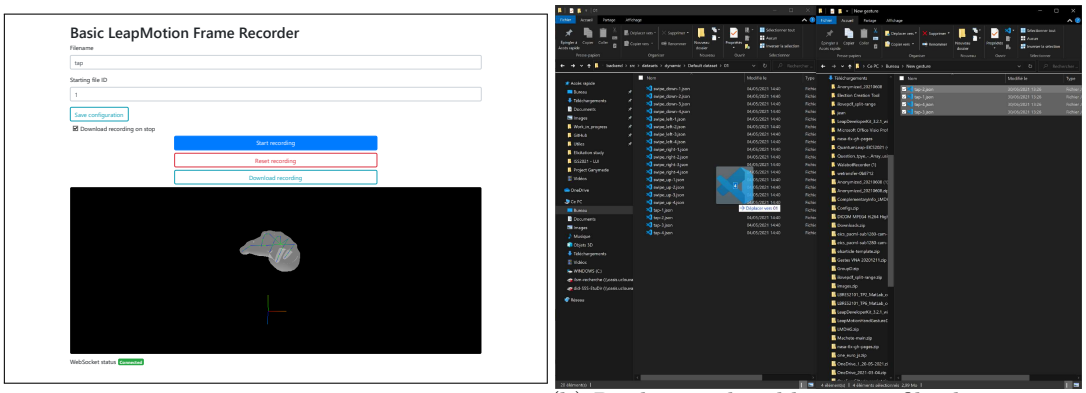
6.1.2. Continuity

LUI enables continuous interaction with media contents. Its built-in segmentation module allows the continuous recognition of gestures while they are being issued, without requiring input from the user, and the static recognizer allows users to manipulate media objects with immediate and continuous feedback.

6.1.3. Customizability

LUI supports customizability by enabling users to include their own gestures (Coyette et al., 2007). Thanks to the modular pipeline architecture, customizing the gesture set is a simple three-step process (Fig. 9) which requires no coding skills, as it does not alter the LUI code:

(1) Record a few templates for the new gesture using a custom software(Fig. 9a).
(2) Overwrite the old gesture files with the new set of templates (Fig. 9b).
(3) Restart the LUI gesture recognition pipeline to propagate the changes, which takes ≤ 10 seconds on a modern computer. This includes the time to load the datasets, train the recognizers on the modifications, connect to the sensor, and re-initialize the segmentation module.



(a) Recording new gesture templates.

(b) Replacing the old gesture files by new templates.

Figure 9.: Illustration of the main steps for customizing the LUI gesture set.

6.2. Limitations

6.2.1. Hardware

The LMC hardware was selected for its numerous benefits, such as its accurate tracking of a large number of hand articulations, its robustness (Weichert et al., 2013), its affordability, and its wide availability. Consequently, the LMC triggered a lot of development in mid-air gesture interaction (Koutsabasis & Vogiatzidakis, 2019). However, it has some limitations, including its sensitivity to lighting conditions and self-occlusion as well as its effective range, which extends from approximately 25 to 600 millimeters above the device (1 inch to 2 feet). These limitations affect the types of gestures that can be recognized, limit the context of use of the system, and restrict end-users' movements. Combining a LMC with a Microsoft Kinect (Marin et al., 2014), a depth sensor (Marin et al., 2016) or combining LMCs (Kiselev et al., 2019) could address the problem of occlusion. (Kiselev et al., 2019) showed how three LMC could be positioned in space to optimize the recognition of 7 hand poses, which are actually finger poses. Recommendations in terms of distance between the devices are given, such as, for example, 32 cm on the XY plane. However, they did not test dynamic gestures. Furthermore, while it is able to track two hands at once, using one LMC for simultaneous multi-user gesture sensing is not recommended as it may reduce motion tracking quality (Ultraleap, 2021) and is unpractical due to the LMC's small interaction zone. Other vision-based sensors such as the MS Kinect and Kinect v2 support a much larger interaction cone, thus enabling multiuser tracking and the recognition of whole-body gestures. However, they only track a small set of joints per hand, which makes them less suited for fine-grained hand gesture recognition compared to the LMC. Radar and wearable sensors may address some of the issues related to vision-based sensors such as sensitivity to lighting conditions, weather, and privacy concerns. However, both suffer from their own shortcomings. Wearable sensors can provide accurate data and allow users to move around freely, but must be worn by the user. Radar sensors are sensitive to background clutter and interference, and many challenges remain regarding their use in gesture recognition (Ahmed et al., 2021).

6.2.2. Software

While the use of simple template-matching recognizers allows users to quickly and easily customize the Lui gesture set with their own gestures, properly trained state-of-the-art machine learning recognizers may be able to accommodate more gesture types and achieve even higher accuracy. Furthermore, taking into account modifications to the gesture set requires a restart of the gesture recognition pipeline. Although this process takes only a few seconds thanks to the use of template-matching recognizers (Brunelli, 2009), as opposed to neural network-based recognizers which may require extensive re-training before recognizing the updated gesture set (Brunelli, 2009), the need for a restart could be avoided altogether. Moreover, the customization process could be further streamlined and simplified by integrating it with the Lui application. This would allow users to modify the gesture set without leaving the application. In addition, although some workarounds were implemented (*e.g.*, rejection thresholds), gesture segmentation is not perfect and parasitic motion may be confused with gestures actually intended by the user. Users thus still need to be careful while interacting with the application to avoid triggering actions unexpectedly. Finally, the current version of the Lui gesture recognition pipeline does not support concurrent interaction from multiple users. As a result, concurrent gestures from two users will interfere with each other.

6.2.3. Miscellaneous

User fatigue may become a problem during long interactions with Lui, as mid-air gestures can be more tiring than traditional interaction methods (Siddhpuria et al., 2017; Jakobsen et al., 2015). In addition, while user-elicited gesture sets have been shown to improve the immediate usability of an application (Wobbrock et al., 2005), the GES procedure followed in this paper is not without flaws. Our choice of illustrations of the referents may have influenced participants' gesture propositions. The small sample size of our study may also not accurately represent the majority of the population, as gesture propositions are affected by participants' cultural background (Archer, 1997). Finally, GES's usually rely on subjective criteria for gesture clustering, which may further impact their findings (Vatavu, 2019). The customizability of Lui contributes to mitigating these issues by enabling end-users to define their own gesture set.

7. Conclusion

This paper presented Lui, an interactive application benefiting from a consistent, continuous, and customizable mid-air gesture interaction for browsing multimedia objects (*i.e.*, photos, videos, documents, and maps) on a large display based on a LMC. We detailed the iterative and participatory process of designing this gesture-based application from the beginning: conducting a contextual gesture elicitation study instead of separate studies for different media, defining new agreement rates and (in-)consistency rates to support designing an intuitive gesture set, implementing a gesture recognition pipeline, performing a comparative testing of selected recognizers on the resulting gesture set to determine the best pipeline configuration, and evaluating the final gestural interface with potential end-users of the system. From this evaluation, we observed that the perceived usability of Lui increased between the two experimental sessions, even if the participants had trouble remembering some of the gestures.

This work has many potential ramifications, which we intend to investigate in future work. First, the current version of Lui relies exclusively on one-handed gestures. While

this choice seems to fit users' preferences, as most gestures proposed in our GES were one-handed, this could change depending on the sensor and the context of use. It would be relevant to implement and evaluate the performance of the system on a gesture set of two-handed, or even whole-body gestures. In addition, we focused on a context in which one user controls the system and other users are spectators. In the future, we plan to investigate multiuser scenarios of two types: (1) a collaborative scenario, where multiple users cooperate on the same screen by performing gestures concurrently, and (2) a "talking stick" scenario where each user has the possibility to interact with the system, but the main control can switch between users (*e.g.*, in a classroom or an office meeting). These scenarios could be achieved by using one sensor capable of sensing multiple users concurrently (*e.g.*, a MS Kinect), or a set of sensors, each capturing gestures from one or more users (*e.g.*, one LMC in front of each seat at a conference table). Next, we will look into incorporating speech recognition into Lui, as it could complement mid-air gestures by providing a simple way to execute complex commands, such as searching for specific content by keywords. Finally, if we seek to deploy the system on public displays, a number of challenges will certainly arise, including dealing with users' privacy concerns and a less forgiving environment. To answer these challenges, we will investigate other types of sensors, such as radar, which may be more appropriate in public scenarios, as they answer most privacy issues related to vision-based sensors and are less sensitive to weather and lighting conditions. UI elements will also have to be carefully designed to attract users, retain their attention, and make them aware of the supported gestures.

Beyond the novel measures defined, we would like to further investigate various forms of agreement among gesture proposals. The agreement rate expresses when pairs of participants agree on a gesture, whereas the co-agreement rate expresses when pairs of participants which agree on a gesture also agree on another. This could be extended by formalizing the various cases when a pair of participants agrees on one gesture proposal, but not on another, or vice versa. In this way, it is expected to define novel measures for expressing these cases as well.

The GitHub repository provides access to the ready-to-deploy/use last version of Lui, along with various modules and files discussed in this paper. We expect to create a Docker version[2] to facilitate this deployment, as this solution enables packaging Lui as a portable container image to run in any environment consistently according to deployment standards.

Acknowledgements

The authors of this paper acknowledge the support of the MIT-Belgium MISTI Program under grant COUHES n°1902675706. Arthur Sluÿters is funded by the "Fonds de la Recherche Scientifique - FNRS" under Grant n°40001931.

References

Ackad, C., Clayphan, A., Tomitsch, M. & Kay, J. (2015). An In-the-Wild Study of Learning Mid-Air Gestures to Browse Hierarchical Information at a Large Interactive Public Display. In *Proceedings of the ACM Int. Joint Conf. on Pervasive and*

[2]See `https://www.docker.com/`

Ubiquitous Computing (UbiComp '15). (pp. 1227-1238). ACM. DOI:https://doi.org/10.1145/2750858.2807532

Ahmed, S., Kallu, K., Ahmed, S. & Cho, S. (2021). Hand Gestures Recognition Using Radar Sensors for Human-Computer-Interaction: A Review. *Remote Sensing*. **13**, 3, 527. DOI:https://www.mdpi.com/2072-4292/13/3/527

Aigner, R., Wigdor, D., Benko, H., Haller, M., Lindbauer, D., Ion, A., Zhao, S. & Koh, J. (2012). Understanding Mid-Air Hand Gestures: A Study of Human Preferences in Usage of Gesture Types for HCI. Microsoft Research, Redmond. https://www.microsoft.com/en-us/research/publication/understanding-mid-air-hand-gestures-a-study-of-human-preferences-in-usage-of-gesture-types-for-hci/

Ali, A., Morris, M. & Wobbrock, J. (2021). "I Am Iron Man": Priming Improves the Learnability and Memorability of User-Elicited Gestures. In *Proceedings of the ACM Int. Conference on Human Factors in Computing Systems (CHI '21)*. (pp. 359:1-359:14). 2021. DOI:https://doi.org/10.1145/3411764.3445758

Andrews, C., Endert, A., Yost, B. & North, C. (2011) Information Visualization on Large, High-Resolution Displays: Issues, Challenges, and Opportunities. *Information Visualization*. **10**, 341-355, DOI:https://doi.org/10.1177/1473871611415997

Annett, M., Ng, A., Dietz, P., Bischof, W., & Gupta, A. (2014). How Low Should We Go? Understanding the Perception of Latency While Inking. In *Proceedings of Graphics Interface (GI '14)*. (pp. 167-174). Canadian Information Processing Society. 2014. DOI:https://graphicsinterface.org/proceedings/gi2014/gi2014-22/

Anthony, L. & Wobbrock, J. (2010). A Lightweight Multistroke Recognizer for User Interface Prototypes. In *Proceedings of Graphics Interface (GI '10)*. (pp. 245-252). DOI:http://dl.acm.org/citation.cfm?id=1839214.1839258

Anthony, L. & Wobbrock, J. (2012). $N-ProTractor: A Fast and Accurate Multistroke Recognizer. In *Proceedings of Graphics Interface (GI '12)*. (pp. 117-120). DOI:http://dl.acm.org/citation.cfm?id=2305276.2305296

Anthony, L., Vatavu, R. & Wobbrock, J. (2013) Understanding the Consistency of Users' Pen and Finger Stroke Gesture Articulation. In *Proceedings of Graphics Interface (GI '13)*. (pp. 87-94). DOI:https://dl.acm.org/doi/10.5555/2532129.2532145

Archer, D. (1997). Unspoken Diversity: Cultural Differences in Gestures. *Qualitative Sociology*. **20**, 79-105. DOI:https://doi.org/10.1023/A:1024716331692

Ardito, C., Buono, P., Costabile, M. & Desolda, G. (2015). Interaction with Large Displays: A Survey. *ACM Computing Surveys*. **47**, *3*, Article 46, 1-38. DOI:https://doi.org/10.1145/2682623

Bachmann, D., Weichert, F. & Rinkenauer, G. (2018). Review of Three-Dimensional Human-Computer Interaction with Focus on the Leap Motion Controller. *Sensors*. **18**, *7*, 2194, DOI:https://doi.org/10.3390/s18072194

Boduch, A. (2019). React Material-UI Cookbook. Packt Publishing. https://www.packtpub.com/application-development/react-material-ui-cookbook

Boulabiar, M., Coppin, G. & Poirier, F. (2014). The Study of the Full Cycle of Gesture Interaction, The Continuum between 2D and 3D. In *Proc. of 16th Int. Conf. on Human-Computer Interaction (HCI International '14)*. Lecture Notes in Computer Science , Vol. 8511. (pp. 24–35). Springer. DOI:10.1007/978-3-319-07230-2_3

Brandon, S. (2014). Mastering Leap Motion. Packt Publishing.

Brunelli, R. (2009). Template Matching Techniques in Computer Vision: Theory and Practice. John Wiley & Sons.

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L. & Vanderdonckt, J. (2003). A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers*. **15**, 289-308. DOI:https://doi.org/10.1016/S0953-5438(03)0

0010-9

Caputo, F. & Giachetti, A. (2015). Evaluation of Basic Object Manipulation Modes for Low-Cost Immersive Virtual Reality. In *Proceedings of 11th Conf. of Italian SIGCHI Chapter*. (pp. 74-77). DOI:https://doi.org/10.1145/2808435.2808439

Caputo, F., Prebianca, P., Carcangiu, A., Spano, L. & Giachetti, A. (2017) A 3 Cent Recognizer: Simple and Effective Retrieval and Classification of Mid-Air Gestures from Single 3D Traces. In *Proc. of Int. Conf. on Smart Tools and Applications in Computer Graphics*. (pp. 9-15). DOI:https://doi.org/10.2312/stag.20171221

Caputo, F., Prebianca, P., Carcangiu, A., Spano, L. & Giachetti, A. (2018). Comparing 3D trajectories for simple mid-air gesture recognition. *Computers & Graphics*. **73**, 17-25. DOI:http://www.sciencedirect.com/science/article/pii/S0097849318300335

Caputo, F., Burato, S., Pavan, G., Voillemin, T., Wannous, H., Vandeborre, J., Maghoumi, M., Taranta II, E., Razmjoo, A., LaViola Jr., J., Manganaro, F., Pini, S., Borghi, G., Vezzani, R., Cucchiara, R., Nguyen, H., Tran, M. & Giachetti, A. (2019). Online Gesture Recognition. In Biasotti, S., Lavoué, G., and Veltkamp, R. (Eds.), *Proc. of Eurographics Workshop on 3D Object Retrieval*. (pp. 93-102). The Eurographics Association. DOI:10.2312/3dor.20191067 . https://diglib.eg.org/handle/10.2312/3dor20191067

Chen, Y., Su, X., Tian, F., Huang, J., Zhang, X., Dai, G., & Wang, H. (2016) Pactolus: A Method for Mid-Air Gesture Segmentation within EMG. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems (CHI '16), Extended Abstracts*. (pp. 1760-1765). ACM. DOI:https://doi.org/10.1145/2851581.2892492

Cheng, H., Yang, L. & Liu, Z. (2016). Survey on 3D Hand Gesture Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*. **26**, 9, 1659-1673. DOI:https://ieeexplore.ieee.org/document/7208833

Cheng, Z. (2017). Recognition and interpretation of multi-touch gesture interaction. PhD thesis. INSA de Rennes, France. 2017 DOI:https://tel.archives-ouvertes.fr/tel-01578068

Colgan, A. How Does the Leap Motion Controller Work? (2017). *Leap Motion Blog*. https://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/

Cook, H., Nguyen, Q., Simoff, S. & Huang, M. (2016). Enabling Gesture Interaction with 3D Point Cloud. In *Proc. of 24th Int. Conf. in Central Europe on Computer Graphics, Visualization And Computer Vision (WSCG '16)*. **2602** pp. 59-68 (2016,6), DOI:https://opus.lib.uts.edu.au/handle/10453/45407

Coyette, A., Schimke, S., Vanderdonckt, J. & Vielhauer, C. (2007). Trainable Sketch Recognizer for Graphical User Interface Design. In *Proc. of IFIP TC 13 Int. Conf. on Human-Computer Interaction (Interact '07)*. Lecture Notes in Computer Sience, Vol. 4662. (pp. 124-135). Springer. DOI:https://link.springer.com/chapter/10.1007/978-3-540-74796-3_14

Delamare, W., Silpasuwanchai, C., Sarcar, S., Shiraki, T. & Ren, X. (2019). On Gesture Combination: An Exploration of a Solution to Augment Gesture Interaction. In *Proc. of ACM Int. Conf. on Interactive Surfaces and Spaces (ISS '19)*. (pp. 135-146). ACM. DOI:https://doi.org/10.1145/3343055.3359706

Dingler, T., Rzayev, R., Shirazi, A. & Henze, N. (2018) Designing Consistent Gestures Across Device Types: Eliciting RSVP Controls for Phone, Watch, and Glasses. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems (CHI '18)*. Paper 419 (pp. 1-12). DOI:https://doi.org/10.1145/3173574.3173993

Dong, H., Figueroa, N. & El Saddik, A. An Elicitation Study on Gesture Attitudes

and Preferences Towards an Interactive Hand-Gesture Vocabulary. *Proceedings of The 23rd ACM International Conference on Multimedia*. pp. 999-1002 (2015), DOI: https://doi.org/10.1145/2733373.2806385

Drossis, G., Grammenos, D., Birliraki, C. & Stephanidis, C. MAGIC: Developing a Multimedia Gallery Supporting mid-Air Gesture-Based Interaction and Control. *Proc. of Int. Conf. on Human-Computer Interaction, Extended Abstracts*. (pp. 303-307), DOI:https://link.springer.com/chapter/10.1007/978-3-642-39473-7_61

Dumas, C., Erdelez, S., Pomerantz, J. & Parthiban, V. Usability testing of LUI: A new human-computer interface for large displays. *Proc. of the Association for Information Science and Technology*. **56**, *1*. (pp. 645-647), DOI:https://doi.org/10.1002/pra2.118

Ekman, P. & Friesen, W. The Repertoire of Nonverbal Behavior: Categories, Origins, Usage, and Coding. *Semiotica*. **1**, 49-98 (1969)

Filho, I., Chen, E., Silva Junior, J. & Silva Barboza, R. Gesture Recognition Using Leap Motion: A Comparison between Machine Learning Algorithms. *ACM SIGGRAPH 2018 Posters*. (2018), DOI:https://doi.org/10.1145/3230744.3230750

Fruchard, B., Lecolinet, E. & Chapuis, O. (2018) How Memorizing Positions or Directions Affects Gesture Learning?. In *Proc. of ACM Int. Conf. on Interactive Surfaces and Spaces (ISS '18)*. (pp. 107-114). ACM. DOI:https://doi.org/10.1145/3279778.3279787

Fruchard, B., Lecolinet, E. & Chapuis, O. (2020, November). Side-Crossing Menus: Enabling Large Sets of Gestures for Small Surfaces. *Proc. ACM Human-Computer Interaction*. **4**. DOI:https://doi.org/10.1145/3427317

Gheran, B., Vanderdonckt, J. & Vatavu, R. (2018) Gestures for Smart Rings: Empirical Results, Insights, and Design Implications. In *Proc. of ACM Int. Conf. on Designing Interactive Systems (DIS '18)*. (pp. 623-635). ACM. DOI:https://doi.org/10.1145/3196709.3196741

Grijincu, D., Nacenta, M. & Kristensson, P. User-Defined Interface Gestures: Dataset and Analysis. *Proc. of 9th ACM Int. Conf. on Interactive Tabletops and Surfaces (ITS '14)*. (pp. 25-34). ACM. DOI:https://doi.org/10.1145/2669485.2669511

Gupta, A., Pietrzak, T., Yau, C., Roussel, N. & Balakrishnan, R. (2017) Summon and Select: Rapid Interaction with Interface Controls in Mid-Air. In *Proceedings of ACM Int. Conf. on Interactive Surfaces and Spaces (ISS '17)*. (pp. 52-61). ACM. DOI: https://doi.org/10.1145/3132272.3134120

Groenewald, C., Anslow, C., Islam, J., Rooney, C., Passmore, P. & Wong, B. Understanding 3D Mid-Air Hand Gestures with Interactive Surfaces and Displays: A Systematic Literature Review. *Proceedings Of The 30th International BCS Human Computer Interaction Conference*. (2016), DOI:https://eprints.mdx.ac.uk/22155/

Guzsvinecz, T., Szucs, V. & Sik-Lanyi, C. (2019). Suitability of the Kinect Sensor and Leap Motion Controller—A Literature Review. *Sensors*. **19**, *5*. Article 1072. DOI: https://www.mdpi.com/1424-8220/19/5/1072

Hinderks, A., Schrepp, M., Domínguez Mayo, F.J., Escalona, M.J., & Thomaschewski, J. (2019, July). Developing a UX KPI based on the user experience questionnaire. *Computer Standards & Interfaces*. **65**, *3*, 38-44. Institute of Mathematical Statistics. DOI:https://doi.org/10.1016/j.csi.2019.01.007

Jakobsen, M., Jansen, Y., Boring, S. & Hornbæk, K. (2015). Should I Stay or Should I Go? Selecting Between Touch and Mid-Air Gestures for Large-Display Interaction. In *Proc. of IFIP TC13 Int. Conf. on Human-Computer Interaction (INTERACT '15)*. Lecture Notes in Computer Science, Vol. 9298 (pp. 455-473). DOI:https://doi.org/10.1007/978-3-319-22698-9_31

Kane, L. & Khanna, P. (2016). A Framework to Plot and Recognize Hand Motion Trajectories towards Development of Non-tactile Interfaces. In *Proc. of the 7th Int. Conf. on Intelligent Human-Computer Interaction (IHCI '15)*, *Procedia Computer Science*. **84**, 6-13, DOI:http://www.sciencedirect.com/science/article/pii/S1877050916300734

Kendall, M. & Babington Smith, B. (1939, September). The Problem of m Rankings. *Annals of Math. Statistics*. **10**, *3*, 275-287. Institute of Mathematical Statistics. DOI: http://www.jstor.org/stable/2235668

Kendon, A. (1988). How gestures can become like words. *Crosscultural Perspectives in Nonverbal Communication*. (pp. 131-141).

Khalaf, A., Alharthi, S., Dolgov, I. & Toups, Z. (2019). A Comparative Study of Hand Gesture Recognition Devices in the Context of Game Design. In *Proc. of ACM Int. Conf. on Interactive Surfaces and Spaces (ISS '19)*. pp. 397-402 (2019), DOI: https://doi.org/10.1145/3343055.3360758

Kiselev, V., Khlamov, M., & Chuvilin, K. (2019). Hand Gesture Recognition with Multiple Leap Motion Devices. In *Proc. of Proceedings of the 24th Conference of Open Innovations Association (FRUCT '24)*. IEEE. DOI:http://10.23919/FRUCT.2019.8711887

Koutsabasis, P. & Domouzis, C. (2016) Mid-Air Browsing and Selection in Image Collections. In *Proc. of ACM Int. Working Conference on Advanced Visual Interfaces (AVI '16)*. (pp. 21-27). ACM. DOI:https://doi.org/10.1145/2909132.2909248

Koutsabasis, P. & Vogiatzidakis, P. (2019, February) Empirical Research in Mid-Air Interaction: A Systematic Review. *Int. J. of Human–Computer Interaction*. **35**, *18*, 1747-1768. DOI:https://doi.org/10.1080/10447318.2019.1572352

Kohlsdorf, D. & Starner, T. (2013, January). MAGIC summoning: towards automatic suggesting and testing of gestures with low probability of false positives during use. *The Journal of Machine Learning Research*. **14**, 209-242. https://www.jmlr.org/papers/volume14/kohlsdorf13a/kohlsdorf13a.pdf

Kratz, S. & Rohs, M. (2010). A $3 Gesture Recognizer: Simple Gesture Recognition for Devices Equipped with 3D Acceleration Sensors. In *Proc. of 15th ACM Int. Conf. on Intelligent User Interfaces (IUI '10)*. (pp. 341-344). ACM. DOI:https://doi.org/10.1145/1719970.1720026

Kratz, S. & Rohs, M. (2011). Protractor3D: A Closed-Form Solution to Rotation-Invariant 3D Gestures. In *Proc. of 16th ACM Int. Conf. on Intelligent User Interfaces (IUI '11)*. (pp. 371-374). ACM. DOI:https://doi.org/10.1145/1943403.1943468

Kratz, S. & Back, M. (2015). Towards Accurate Automatic Segmentation of IMU-Tracked Motion Gestures. In *Proc. of 33rd ACM Int. Conf. on Human Factors in Computing Systems (CHI '15), Extended Abstracts*. (pp. 1337-1342). ACM. DOI: https://doi.org/10.1145/2702613.2732922

LaViola, J. (2013) 3D Gestural Interaction: The State of the Field. (2013) *International Scholarly Research Notices*. **V**ol. 2013, Article ID 514641. DOI:https://www.hindawi.com/journals/isrn/2013/514641/

Lee, J., Olwal, A., Ishii, H. & Boulanger, C. (2013). SpaceTop: Integrating 2D and Spatial 3D Interactions in a See-through Desktop Environment. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems (CHI '13)*. (pp. 189-192). ACM. DOI:https://doi.org/10.1145/2470654.2470680

Li, Y. (2010). Protractor: A Fast and Accurate Gesture Recognizer. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems (CHI '10)*. (pp. 2169-2172). ACM. DOI:http://doi.acm.org/10.1145/1753326.1753654

López, G., Quesada, L. & Guerrero, L. (2015) A Gesture-Based Interaction Approach

for Manipulating Augmented Objects Using Leap Motion. In *Proc. of 7th Int. Work-Conference on Ambient Assisted Living (IWAAL '15), ICT-based Solutions in Real Life Situations*. Lecture Notes in Computer Science, Vol. 9455 (pp. 231-243). Springer. DOI:https://link.springer.com/chapter/10.1007/978-3-319-26410-3_22

Maghoumi, M. & LaViola, J. (2019). DeepGRU: Deep Gesture Recognition Utility. In *Proc. of 14th Int. Symposium on Visual Computing (ISVC '19), Advances in Visual Computing*. Lecture Notes in Computer Science, Vol. 11844 (pp. 16-31). Springer. DOI:https://link.springer.com/chapter/10.1007/978-3-030-33720-9_2

Magrofuoco, N., Roselli, P. & Vanderdonckt, J. (2021) Two-Dimensional Stroke Gesture Recognition: A Survey. *ACM Computing Surveys*. **54**, 7. Article 155. 1-36. DOI: https://doi.org/10.1145/3465400

Marin, G., Dominio, F. & Zanuttigh, P. (2014) Hand gesture recognition with Leap Motion and Kinect devices. *2014 IEEE International Conference On Image Processing, ICIP 2014*. pp. 1565-1569. DOI:https://doi.org/10.1109/ICIP.2014.7025313

Marin, G., Dominio, F. & Zanuttigh, P. (2016, Nov.). Hand Gesture Recognition with Jointly Calibrated Leap Motion and Depth Sensor. *Multimedia Tools And Applications*. **75**, 14991-15015. DOI:https://doi.org/10.1007/s11042-015-2451-6

McNeill, D. (1995). Hand and Mind: What Gestures Reveal About Thought. The University of Chicago Press.

Morris, M., Wobbrock, J. & Wilson, A. (2010) Understanding Users' Preferences for Surface Gestures. In *Proceedings of Graphics Interface (GI' 10)*. (pp. 261-268). DOI: https://dl.acm.org/doi/10.5555/1839214.1839260

Morris, M. (2012). Web on the Wall: Insights from a Multimodal Interaction Elicitation Study. In *Proc. of ACM Int. Conf. on Interactive Tabletops and Surfaces (ITS '12)*. (pp. 95-104). ACM. DOI:https://doi.org/10.1145/2396636.2396651

Morris, M., Danielescu, A., Drucker, S., Fisher, D., Lee, B., Schraefel, M. & Wobbrock, J. (2014, May) Reducing Legacy Bias in Gesture Elicitation Studies. *Interactions*. **21**, 40-45. DOI:https://doi.org/10.1145/2591689

Nacenta, M., Kamber, Y., Qiang, Y. & Kristensson, P. (2013). Memorability of Pre-Designed and User-Defined Gesture Sets. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems (CHI '13)*. (pp. 1099-1108). ACM. DOI:https://doi.org/10.1145/2470654.2466142

Nancel, M., Wagner, J., Pietriga, E., Chapuis, O. & Mackay, W. (2011) Mid-Air Pan-and-Zoom on Wall-Sized Displays. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems (CHI '11)*. (pp. 177-186). ACM. DOI:https://doi.org/10.1145/1978942.1978969

Nielsen, J. (1994). Usability Engineering. Elsevier Science, Amsterdam. DOI:https://books.google.be/books?id=95As2OF67f0C

Ortega, F., Galvan, A., Tarre, K., Barreto, A., Rishe, N., Bernal, J., Balcazar, R. & Thomas, J. (2017). Gesture elicitation for 3D travel via multi-touch and mid-Air systems for procedurally generated pseudo-universe. In *Proc. of IEEE Int. Symposium on 3D User Interfaces (3DUI'17)*. (pp. 144-153). IEEE. DOI:https://doi.org/10.1109/3DUI.2017.7893331

Ousmer, M., Sluÿters, A., Magrofuoco, N., Roselli, P., & Vanderdonckt, J. (2020). Recognizing 3D Trajectories as 2D Multi-stroke Gestures. *Proc. ACM Hum. Comput. Interact. (ISS)*, 4. (pp. 198:1-198:2). ACM Press. DOI:https://doi.org/10.1145/3427326

Park, C., Cho, H., Park, S., Yoon, Y. & Jung, S. (2019) HandPoseMenu: Hand Posture-Based Virtual Menus for Changing Interaction Mode in 3D Space. In *Proc. of ACM*

Int. Conf. on Interactive Surfaces and Spaces (ITS '19). (pp. 361-366). ACM. DOI: https://doi.org/10.1145/3343055.3360752

Parthiban, V. & Lee, A.J. (2019). LUI: A multimodal, intelligent interface for large displays. In *Proc. of the 17th International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI '19)*. (pp. 48:1–48:2). ACM Press, New York. DOI:https://doi.org/10.1145/3359997.3365743

Piumsomboon, T., Clark, A., Billinghurst, M. & Cockburn, A. (2013). User-Defined Gestures for Augmented Reality. In *Proc. of IFIP TC13 Int. Conf. on Human-Computer Interaction (INTERACT '13)*. Lecture Notes in Computer Science, Vol. 8118 (pp. 282-299). DOI:https://doi.org/10.1007/978-3-642-40480-1_18

Pham, T., Vermeulen, J., Tang, A. & MacDonald Vermeulen, L. (2018). Scale Impacts Elicited Gestures for Manipulating Holograms: Implications for AR Gesture Design. In *Proc. of ACM Int. Conf. on Designing Interactive Systems (DIS '18)*. (pp. 227-240). ACM. DOI:https://doi.org/10.1145/3196709.3196719

Rajabiyazdi, F., Walny, J., Mah, C., Brosz, J. & Carpendale, S. (2015). Understanding Researchers' Use of a Large, High-Resolution Display Across Disciplines. In *Proc. of ACM Int. Conf. on Interactive Tabletops and Surfaces (ITS '15)*. (pp. 107-116). ACM. DOI:https://doi.org/10.1145/2817721.2817735

Rose, S., Clark, M., Samouel, P. & Hair, N. (2012). Online Customer Experience in e-Retailing: An empirical model of Antecedents and Outcomes. In *Journal Of Retailing*. **88**, (pp. 308-322). DOI:https://doi.org/10.1016/j.jretai.2012.03.001

Ruiz, J., Li, Y. & Lank, E. (2011). User-defined Motion Gestures for Mobile Interaction. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems (CHI '11)*. (pp. 197-206). ACM. DOI:http://doi.acm.org/10.1145/1978942.1978971

Schrepp, M., Thomaschewsk, J., & Hinderks, A. (June 2017). Construction of a Benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*, **4**, *4*, 197-214. DOI:https://doi.org/10.9781/ijimai.2017.445

Schrepp, M. & Jörg, T (December 2019). Design and validation of a framework for the creation of User Experience Questionnaires. *International Journal Of Interactive Multimedia And Artificial Intelligence*. **5**, 88-95. DOI:https://doi.org/10.9781/ijimai.2019.06.006

Siddhpuria, S., Katsuragawa, K., Wallace, J. & Lank, E. (2017). Exploring At-Your-Side Gestural Interaction for Ubiquitous Environments. In *Proc. of ACM Int. Conf. on Designing Interactive Systems (DIS '17)*. (pp. 1111-1122). ACM. DOI:https://doi.org/10.1145/3064663.3064695

Spiegelmock, M. (2013). Leap Motion Development Essentials. Packt Publishing. https://www.packtpub.com/eu/hardware-and-creative/leap-motion-development-essentials

Steinmetz, R. & Nahrstedt, K. (2004). Multimedia Applications. In *Multimedia Applications*. (pp. 197-214). DOI:https://doi.org/10.1007/978-3-662-08876-0_9

Taranta II, E., Samiei, A., Maghoumi, M., Khaloo, P., Pittman, C. & LaViola Jr., J. (2017). Jackknife: A Reliable Recognizer with Few Samples and Many Modalities. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems (CHI '17)*. (pp. 5850-5861). ACM. DOI:http://doi.acm.org/10.1145/3025453.3026002

Taranta II, E., Pittman, C., Maghoumi, M., Maslych, M., Moolenaar, Y. & LaViola Jr, J. (2021, January). Machete: Easy, Efficient, and Precise Continuous Custom Gesture Segmentation. *ACM Transactions on Computer-Human Interaction*. **28**, *1*. Article 5, 1-45. DOI:https://doi.org/10.1145/3428068

Ultraleap API Overview. (2021). *Leap Motion Blog*, https://developer-archive.le

apmotion.com/documentation/csharp/devguide/Leap_Overview.html, [Online; accessed 02-October-2021]

Vanderdonckt, J., Roselli, P. & Pérez-Medina, J. (2018). !FTL, an Articulation-Invariant Stroke Gesture Recognizer with Controllable Position, Scale, and Rotation Invariances. In *Proc. of 20th ACM Int. Conf. on Multimodal Interaction (ICMI '18)*. (pp. 125-134). ACM. DOI:https://doi.org/10.1145/3242969.3243032

Vatavu, R. (2011). The Effect of Sampling Rate on the Performance of Template-Based Gesture Recognizers. In *Proc. of 13th Int. Conf. on Multimodal Interfaces (ICMI '11)*. (pp. 271-278). ACM. DOI:https://doi.org/10.1145/2070481.2070531

Vatavu, R., Anthony, L. & Wobbrock, J. (2012). Gestures as Point Clouds: A $P Recognizer for User Interface Prototypes. In *Proc. of 14th ACM Int. Conf. on Multimodal Interaction (ICMI '12)*. (pp. 273-280). ACM. DOI:http://doi.acm.org/10.1145/2388676.2388732

Vatavu, R. (2013). The impact of motion dimensionality and bit cardinality on the design of 3D gesture recognizers. *International Journal of Human-Computer Studies*. **71**, 387-409. DOI:http://www.sciencedirect.com/science/article/pii/S1071581912002108

Vatavu, R. & Zaiti, I. (2014). Leap Gestures for TV: Insights from an Elicitation Study. In *Proc. of The ACM Int. Conf. on Interactive Experiences for TV and Online Video*, (pp. 131-138). ACM. DOI:https://doi.org/10.1145/2602299.2602316

Vatavu, R. & Wobbrock, J. (2015). Formalizing Agreement Analysis for Elicitation Studies: New Measures, Significance Test, and Toolkit. In *Proc. of 33rd ACM Int. Conf. on Human Factors in Computing Systems (CHI '15)*. (pp. 1325-1334). ACM. DOI:https://doi.org/10.1145/2702123.2702223

Vatavu, R. (2017). Improving Gesture Recognition Accuracy on Touch Screens for Users with Low Vision. In *Proc. of ACM Conference on Human Factors in Computing Systems (CHI '17)*. pp. 4667-4679). DOI:https://doi.org/10.1145/3025453.3025941

Vatavu, R., Anthony, L. & Wobbrock, J. (2018). $Q: A Super-quick, Articulation-invariant Stroke-gesture Recognizer for Low-resource Devices. In *Proc. of 20th Int. Conf. on Human-Computer Interaction With Mobile Devices and Services (MobileHCI '18)*. (pp. 23:1-23:12). ACM. DOI:http://doi.acm.org/10.1145/3229434.3229465

Vatavu, R. The Dissimilarity-Consensus Approach to Agreement Analysis in Gesture Elicitation Studies. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems*. (pp. 1-13). ACM. DOI:https://doi.org/10.1145/3290605.3300454

Villarreal-Narvaez, S., Vanderdonckt, J., Vatavu, R. & Wobbrock, J. (2020). A Systematic Review of Gesture Elicitation Studies: What Can We Learn from 216 Studies?. In *Proc. of ACM Int. Conf. on Designing Interactive Systems (DIS '20)*. (pp. 855-872). ACM. DOI:https://doi.org/10.1145/3357236.3395511 https://www.zotero.org/groups/2132650/gesture_elicitation_studies.

Vlaming, L., Smit, J. & Isenberg, T. (2008). Presenting using two-handed interaction in open space. In *Proc. of 3rd IEEE Int. Workshop on Horizontal Interactive Human Computer Systems*. (pp. 29-32). IEEE.

Vogel, D. & Balakrishnan, R. (2005). Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proc. of 18th ACM Symposium on User Interface Software and Technology (UIST '05)*. (pp. 33-42). ACM. DOI:https://doi.org/10.1145/1095034.1095041

Vogiatzidakis, P. & Koutsabasis, P. (2019, June). *Frame-Based Elicitation of Mid-Air Gestures for a Smart Home Device Ecosystem*. **6**, *2*, 23. DOI:https://doi.org/10

.3390/informatics6020023

Vogiatzidakis, P. & Koutsabasis, P. (2020, August). Mid-Air Gesture Control of Multiple Home Devices in Spatial Augmented Reality Prototype. *Multimodal Technologies and Interaction*. **4**, *3*, 61. DOI:https://www.mdpi.com/2414-4088/4/3/61

Vogiatzidakis, P. & Koutsabasis, P. (2022, March). 'Address and Command': Two-Handed Mid-Air Interactions with Multiple Home Devices . *International Journal Of Human-Computer Studies*. **159**. DOI:https://doi.org/10.1016/j.ijhcs.2021.102755

Weichert, F., Bachmann, D., Rudak, B. & Fisseler, D. (2013, May). Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors*. **13**, *5*, 6380-6393. DOI:https://www.mdpi.com/1424-8220/13/5/6380

Wittorf, M. & Jakobsen, M. (2016). Eliciting Mid-Air Gestures for Wall-Display Interaction. In *Proc. of 9th Nordic Conf. on Human-Computer Interaction (NordiCHI '16)*. (pp. 1-4). ACM. DOI:https://doi.org/10.1145/2971485.2971503

Wobbrock, J., Aung, H., Rothrock, B. & Myers, B. (2005). Maximizing the Guessability of Symbolic Input. In *Proc. of Int. Conf. on Human Factors in Computing Systems, Extended Abstracts (CHI '05)*. (pp. 1869-1872). ACM. DOI:https://doi.org/10.1145/1056808.1057043

Wobbrock, J., Wilson, A. & Li, Y. (2007). Gestures Without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In *Proc. of 20th ACM Symposium on User Interface Software and Technology (UIST '07)*. (pp. 159-168). ACM. DOI:http://doi.acm.org/10.1145/1294211.1294238

Wobbrock, J., Morris, M. & Wilson, A. (2009). User-defined Gestures for Surface Computing. In *Proc. of ACM Int. Conf. on Human Factors in Computing Systems (CHI '09)*. (pp. 1083-1092). ACM. DOI:http://doi.acm.org/10.1145/1518701.1518866

Yasen, M. & Jusoh, S. (2019, September). A systematic review on hand gesture recognition techniques, challenges and applications. *PeerJ Computer Science*. **5** pp. e218. DOI:https://doi.org/10.7717/peerj-cs.218

Yoo, S., Parker, C., Kay, J. & Tomitsch, M. (2015). To Dwell or Not to Dwell: An Evaluation of Mid-Air Gestures for Large Information Displays. In *Proc. of Annual Meeting of The Australian Special Interest Group for Computer Human Interaction*. (pp. 187-191). DOI:https://doi.org/10.1145/2838739.2838819

Zhai, S., Kristensson, P., Appert, C., Anderson, T. & Cao, X. (2012). Foundational Issues in Touch-Surface Stroke Gesture Design — An Integrative Review. *Foundations And Trends in Human–Computer Interaction*. **5**, 97-205. DOI:http://dx.doi.org/10.1561/1100000012

Zigelbaum, J., Browning, A., Leithinger, D., Bau, O. & Ishii, H. (2010). G-Stalt: A Chirocentric, Spatiotemporal, and Telekinetic Gestural Interface. In *Proc. of 4th Int. Conf. on Tangible, Embedded, and Embodied Interaction (TEI '10)*. (pp. 261-264). ACM. DOI:https://doi.org/10.1145/1709886.1709939

Appendix A. Contextual Gesture Elicitation Study

This appendix summarizes the contextual gesture elicitation study conducted to elicit mid-air gestures for multiple media simultaneously (Section 4.1).

Participants. 23 participants (6 females), aged between 18 and 56 years old (M=27.1, SD=10.2 years), volunteered for the study. Their occupations include students and employees, mostly in IT, art, and engineering. All but two of the participants never used an LMC, but all of them regularly used a computer. Only one participant did not frequently use a smartphone. 95.7% of the participants (22/23) were right-handed.

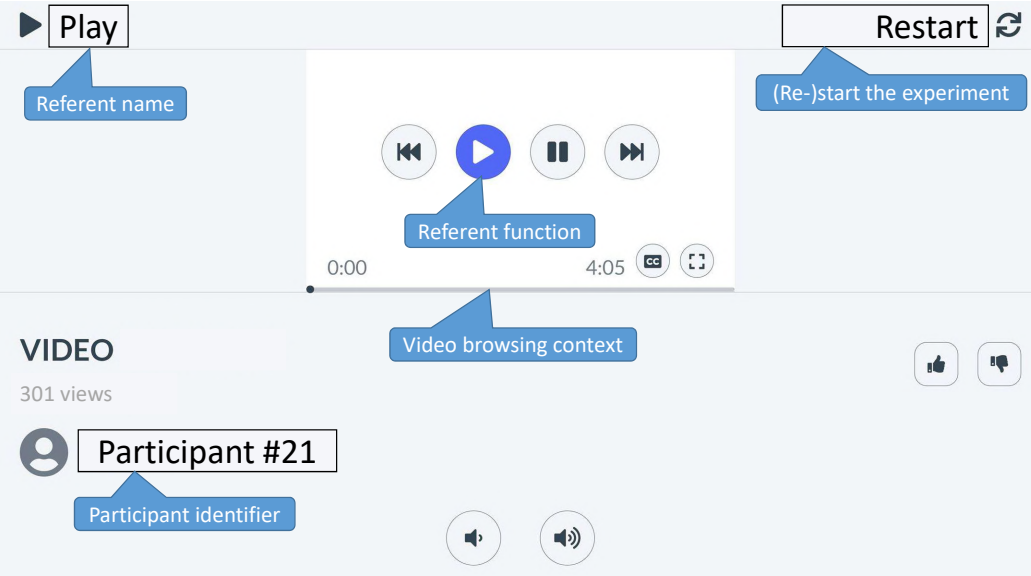


Figure A1.: Screenshot of the software for contextual gesture elicitation.

Apparatus. The experiment involved three devices: the LMC to capture participants' mid-air gesture proposals, a smartphone to record their touch gesture proposals, and a large screen to display the referents. A laptop controlled the display and recorded the LMC gestures using a custom software suited for contextual gesture elicitation.

Design. The experiment manipulates one main independent variable:

REFERENT: a within-subject nominal variable with 18 conditions. Each condition represents a common task to perform in multimedia environments, such as in a music player, a video gallery or a PowerPoint presentation: (1) previous; (2) next; (3) enable fullscreen; (4) disable fullscreen; (5) zoom in; (6) zoom out; (7) volume up; (8) volume down; (9) rotate 90° clockwise; (10) rotate 90° anti-clockwise; (11) play; (12) pause; (13) like; (14) dislike; (15) fast forward 5 seconds; (16) rewind 5 seconds; (17) enable subtitles; and (18) disable subtitles. The number of conditions has been purposefully kept small by combining similar referents for different contexts together, *e.g.*, there is no distinction between going to the next picture and going to the next video.

The following measures were employed to evaluate and understand users' preferences for gestures captured by the LMC (Gheran et al., 2018):

(1) The agreement rate $AR(r)$ was computed for each REFERENT.
(2) The THINKING-TIME measures the time, in seconds, needed by participants to propose a gesture for a given referent.
(3) The GOODNESS-OF-FIT represents participants' subjective assessment, as a rating between 1 and 5, of their confidence about how well a gesture fits a referent.

Task. Participants were installed at a desk and faced a large screen situated at a few meters on the opposite side of the desk. The LMC was positioned in front of them, and a smartphone was placed on their right. Each participant was presented with a series of referents in a randomized order. For each referent, participants were asked to propose a mid-air gesture that fits the referent, in an order determined randomly. Participants' thinking time was measured for each gesture proposal. No constraint was imposed on the participants' choices. After each gesture proposal, the participants estimated its goodness-of-fit on a Likert scale from 1 (very unsatisfied) to 5 (very satisfied).

Results. 414 mid-air gesture proposals were collected from 23 (participants) × 18 (referents) conditions. These gesture proposals were then clustered into 112 groups of similar gestures. For this clustering, we distinguished between gestures according to their direction (*i.e.*, similar gestures performed in different directions are considered different) but grouped similar gestures performed with different poses together (*e.g.*, swiping with a flat hand is considered the same as swiping while pointing the index finger). Table A1 summarizes the most agreed upon gestures for each referent.

Referent	AR (magnitude)	First most agreed gesture	Second most agreed gesture
1. *Previous*	.498 (H)	Flick right	Flick left
2. *Next*	.498 (H)	Flick left	Flick right
3. *Enable fullscreen*	.273 (M)	Bring hands further apart	Pinch out
4. *Disable fullscreen*	.146 (M)	Bring hands closer	Pinch in
5. *Zoom in*	.320 (H)	Pinch out	Bring hands further apart
6. *Zoom out*	.300 (M)	Pinch in	Bring hands closer
7. *Volume up*	.478 (H)	Swipe up	Rotate a knob clockwise
8. *Volume down*	.316 (H)	Swipe down	Rotate a knob anti-clockwise
9. *Rotate 90° clockwise*	.830 (V)	Rotate a knob clockwise	N/A
10. *Rotate 90° anti-clockwise*	.830 (V)	Rotate a knob anti-clockwise	N/A
11. *Play*	.190 (M)	Tap with the index finger	Draw the "play" symbol
12. *Pause*	.166 (M)	Tap with the index finger	Swipe down with two fingers
13. *Like*	.751 (V)	Thumb up	N/A
14. *Dislike*	.751 (V)	Thumbs down	N/A
15. *Fast-forward 5 seconds*	.087 (L)	Flick right twice	Swipe left
16. *Rewind 5 seconds*	.095 (L)	Swipe right	Flick left twice
17. *Enable subtitles*	.032 (L)	Swipe up	Push
18. *Disable subtitles*	.028 (L)	Swipe down	Pull

Table A1.: First and second most agreed gesture proposals for each referent. Referents above average agreement rate are highlighted. Magnitude of the Agreement Rate (AR) (Vatavu & Wobbrock, 2015): L=low, M=medium, H=high, V=very high.

Overall, agreement rates are quite high, between .028 and .83 (M=.366, SD=.268). These results fit within the reported agreement rates in the literature of gesture elicitation; see (Vatavu & Wobbrock, 2015) (p. 1332) that summarize agreement rates of 18 studies. According to the recommendations of Vatavu & Wobbrock (2015) to interpret the magnitudes of agreement rates, most of our results fall inside the high consensus (.300−.500) category. These results were confirmed by Kendall's coefficient of concordance for estimating the inter-rater reliability: W=.564 ($\chi^2(26)$=220.454, p<.0001, large effect). We computed Cronbach's α=0.875 and Guttman's λ_2=.989 (with 2,000 iteration) coefficients, which indicate that the internal consistency was rather reliable among participants of this contextual GES.

Gestures' Goodness of Fit. Participants rated their gesture proposals with numbers from 1 (poor fit) to 5 (excellent fit) to denote their confidence in the goodness of fit of their proposals. Overall, participants were satisfied with their gesture propositions

as shown by the high average goodness of fit (M=4.21, SD=0.88). Goodness-of-Fit correlated significantly with the Agreement-Rate (Pearson's $r_{(N=18)}$=.711, R^2=.505, p=.00094): referents that reached high agreement rates were assigned gestures that were rated a good fit (Figure A2).

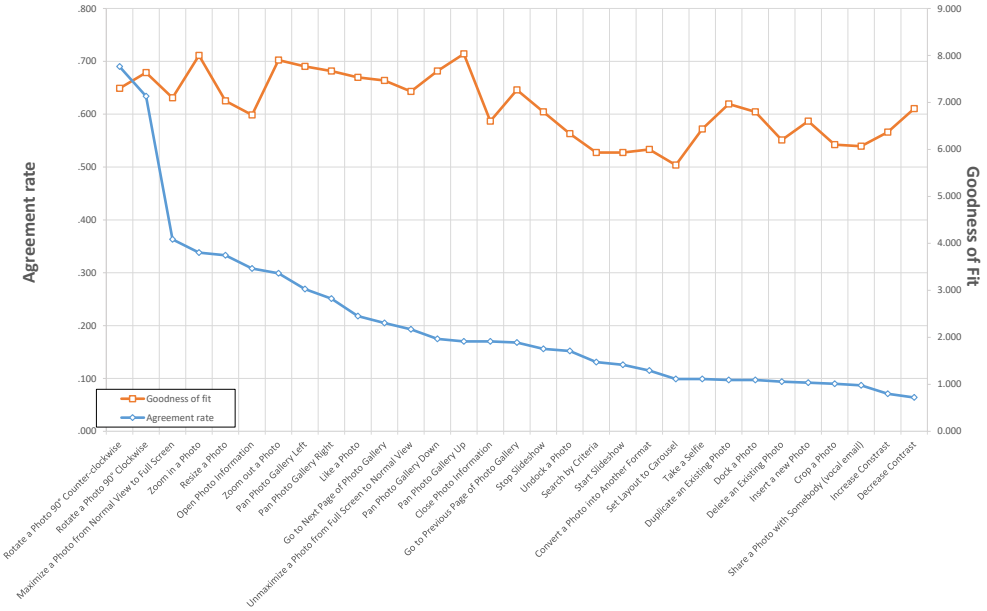


Figure A2.: Relationship between Agreement-Rate and Goodness-of-Fit.

Thinking Time. The average thinking time was usually quite low (M=6.15 s, SD=9.12 s). Thinking-Time correlated significantly with Agreement-Rate (Pearson's $r_{(N=18)}$= − .620, R^2=.384, p=.006) (Fig. A3): the agreement rate decreased for longer thinking times. A few referents showed considerably higher than average thinking times: disable subtitles, enable subtitles, and skip backward 5 seconds. These referents represent tasks that participants are less likely to perform regularly on their devices.

Consistency across Resulting Gestures. To address consistency across gestures resulting from the elicitation study, we followed a method aimed at categorizing similar functions for various media types and keeping the same gesture across various media functions. Fig. A4 depicts for each cell the gesture class identifier (1,...,112) assigned to each referent for all participants (P1,...,P23) with a color representing the media that is randomly selected for each participant: orange for photos, yellow for videos, grey for maps, and blue for documents. Each line of Fig. A4 covers the media suitable for the intended function. For example, Fullscreen On is applicable for all four media types, while Subtitles off is only valid for videos.

The rightmost column of Fig. A4 reproduces the agreement rate $AR(r)$, applicable for all participants, all media by referent (which is referred to as inter-participant, inter-referent, inter-media rate), along with their magnitude (*i.e.*, low, medium, high, very high). The maximum rate is obtained for *AR(Rotate clockwise)*=*AR(Rotate counterclockwise)*=.830 with a very high magnitude, while the minimum rate is obtained for *AR(Subtitles off)*=.028 with a low magnitude. Overall, an average agreement rate *AR(All)*=.366 is obtained with a high magnitude, which is an encouraging result. This average rate therefore expresses the agreement for all participants, referents, and media (inter-participant, inter-referent, inter-media).

The leftmost column of Fig. A4 reproduces the co-agreement rate between pairs of

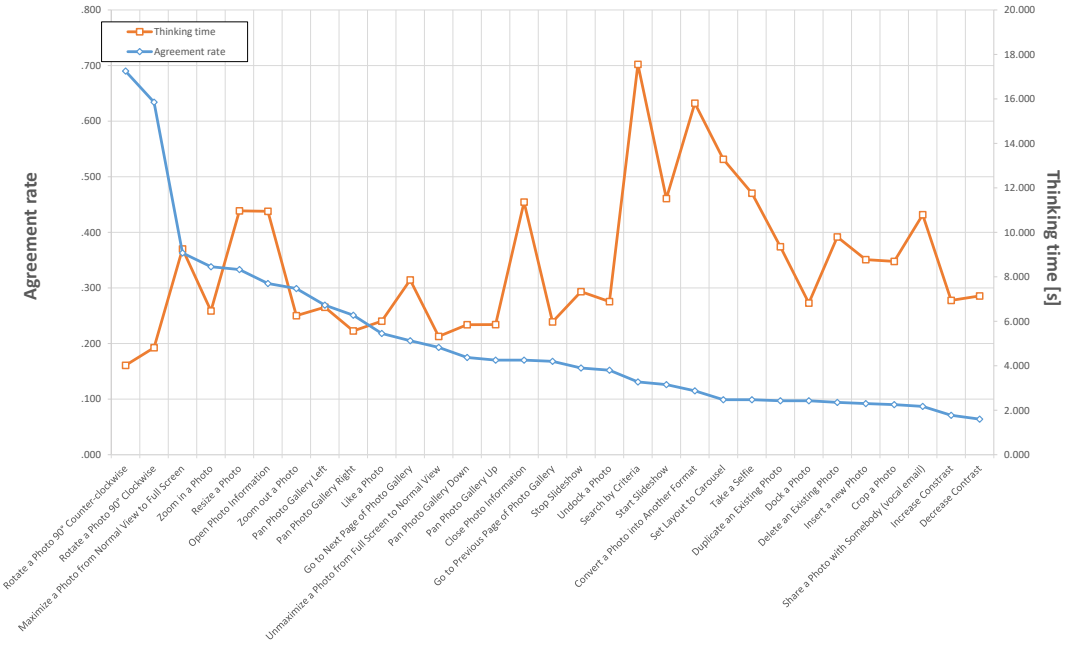


Figure A3.: Relationship between AGREEMENT-RATE and THINKING-TIME.

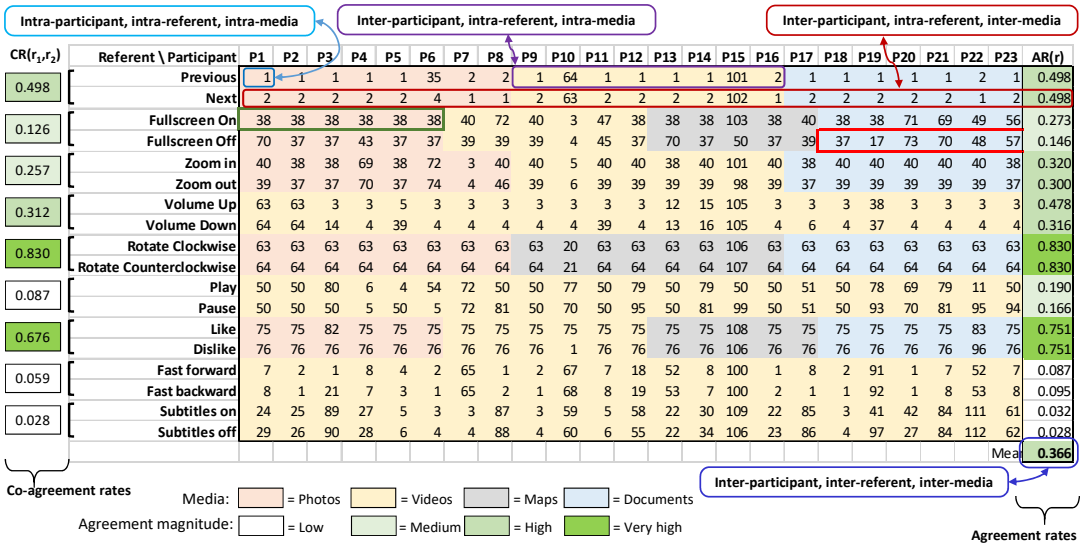
related referents. The maximum co-agreement rate is obtained for *CR(Rotate clockwise, Rotate counterclockwise)*=.830, again with a very high magnitude. The minimum co-agreement rate is *CR(Subtitle on, Subtitles off)*=.028, with a low magnitude.

Measures for Media Agreement and Consistency. The agreement rate reported in the rightmost column in Fig. A4 expresses for each line the inter-participant, intra-referent, inter-media agreement. As this measure is merged for all media, another measure $AR(r_{i\,\text{media}})$ needs to express, for a given *media* (*i.e.*, photos, videos, maps, and documents), the agreement among the media-specific gestures elicited for a referent (intra-referent in Fig. A4). In addition, I, the inconsistency among these agreement rates for all media-specific referents is computed as follows:

$$AR(r_{i\,\text{media}}) = \frac{\sum\limits_{P_j \subseteq P_{i\,\text{media}}} \frac{1}{2}|P_j|\,(|P_j|-1)}{\frac{1}{2}|P_j|\,(|P_j|-1)}$$

$$I = \frac{1}{AR(r_{max\,\text{media}})}\sqrt{\sum_{j=1}^{i}\left(AR(r_{j\,\text{media}}) - \overline{AR(r_{m\,\text{media}})}\right)^2}$$

where *media* $\in$ {photos, videos, maps, document}, $r_{i\,\text{media}}$ denotes the referent for which the i media-specific gestures are elicited, $P_{i\,\text{media}}$ denotes the set of i media-specific elicited gestures, $|P_j|$ denotes the number of gestures elicited for the j^{th} subgroup of $P_{i\,\text{media}}$, $AR(r_{max\,\text{media}})$ denotes the maximum agreement rate obtained for a media-specific referent, and $\overline{AR(r_{m\,\text{media}})}$ denotes the average agreement rate obtained for a media-specific referent $r_{i\,\text{media}}$. The upper part of the above equation particularizes the general formula for agreement rate specifically for one media at a time. The bottom part of the equation computes all deviations of the media-specific agreements independently of their frequency and normalizes it with respect to its



$CR(r_1,r_2)$	Referent \ Participant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	AR(r)
0.498	Previous	1	1	1	1	1	35	2	2	1	64	1	1	1	1	101	2	1	1	1	1	1	2	1	0.498
	Next	2	2	2	2	2	4	1	1	2	63	2	2	2	2	102	1	2	2	2	2	2	1	2	0.498
0.126	Fullscreen On	38	38	38	38	38	38	40	72	40	3	47	38	38	38	103	38	40	38	38	71	69	49	56	0.273
	Fullscreen Off	70	37	37	43	37	37	39	39	39	4	45	37	70	37	50	37	39	37	17	73	70	48	57	0.146
0.257	Zoom in	40	38	38	69	38	72	3	40	40	5	40	40	38	40	101	40	38	40	40	40	40	40	38	0.320
	Zoom out	39	37	37	70	37	74	4	46	39	6	39	39	39	39	98	39	37	39	39	39	39	39	37	0.300
0.312	Volume Up	63	63	3	3	5	3	3	3	3	3	3	3	12	15	105	3	3	3	38	3	3	3	3	0.478
	Volume Down	64	64	14	4	39	4	4	4	4	4	39	4	13	16	105	4	6	4	37	4	4	4	4	0.316
0.830	Rotate Clockwise	63	63	63	63	63	63	63	63	63	20	63	63	63	63	106	63	63	63	63	63	63	63	63	0.830
	Rotate Counterclockwise	64	64	64	64	64	64	64	64	64	21	64	64	64	64	107	64	64	64	64	64	64	64	64	0.830
0.087	Play	50	50	80	6	4	54	72	50	50	77	50	79	50	79	50	50	51	50	78	69	79	11	50	0.190
	Pause	50	50	50	5	50	5	72	81	50	70	50	95	50	81	99	50	51	50	93	70	81	95	94	0.166
0.676	Like	75	75	82	75	75	75	75	75	75	75	75	75	75	75	108	75	75	75	75	75	75	83	75	0.751
	Dislike	76	76	76	76	76	76	76	76	76	1	76	76	76	76	106	76	76	76	76	76	76	96	76	0.751
0.059	Fast forward	7	2	1	8	4	2	65	1	2	67	7	18	52	8	100	1	8	2	91	1	7	52	7	0.087
	Fast backward	8	1	21	7	3	1	65	2	1	68	8	19	53	7	100	2	1	1	92	1	8	53	8	0.095
0.028	Subtitles on	24	25	89	27	5	3	3	87	3	59	5	58	22	30	109	22	85	3	41	42	84	111	61	0.032
	Subtitles off	29	26	90	28	6	4	4	88	4	60	6	55	22	34	106	23	86	4	97	27	84	112	62	0.028
																								Mean	0.366

Figure A4.: (Co-)agreement rates obtained by a contextual GES. The leftmost part reports the co-agreement between pairs of related referents, while the rightmost part reports the agreement for each individual referent. This rate is computed independently of the media types randomly selected for each participant.

	Photos	Videos	Docum	Maps	Inconsist.	Consistency	Consist %
Previous	0.393	0.357	0.714		0.225	0.775	78%
Next	0.393	0.357	0.714		0.225	0.775	78%
Fullscreen On	1.000	0.067	0.067	0.300	0.383	0.617	62%
Fullscreen Off	0.400	0.200	0.000	0.100	0.370	0.630	63%
Zoom in	0.143	0.357	0.524		0.298	0.702	70%
Zoom out	0.107	0.536	0.524		0.372	0.628	63%
Volume Up		0.478			0.000	1.000	100%
Volume Down		0.316			0.000	1.000	100%
Rotate Clockwise	1.000		1.000	0.536	0.219	0.781	78%
Rotate Counterclockwise	1.000		1.000	0.536	0.219	0.781	78%
Play		0.190			0.000	1.000	100%
Pause		0.166			0.000	1.000	100%
Like	0.667	1.000	0.667	0.600	0.156	0.844	84%
Dislike	1.000	0.667	0.667	0.600	0.156	0.844	84%
Fast forward		0.087			0.000	1.000	100%
Fast backward		0.095			0.000	1.000	100%
Subtitles on		0.032			0.000	1.000	100%
Subtitles off		0.028			0.000	1.000	100%
Inconsistency	0.349	0.256	0.318	0.308			
Consistency	0.651	0.744	0.682	0.692			
Consist %	65%	74%	68%	69%			

Inter-participant, inter-referent, intra-media

Inter-participant, intra-referent, inter-media

Figure A5.: Inter-media and intra-media agreement and (in)consistency rates.

maximum. In this way, the inconsistency rate is not bound to how many media types have been used for a particular function (for example, 8 participants elicited a gesture for the "Previous" function for photos, 8 for videos, and 7 for documents in Fig. A4.

The *inconsistency rate* I expresses to what extent there is any discrepancy between agreement rates in two conditions: among all referents for a given media (columns in Fig. A5) and among all media for a given referent (rows in Fig. A5). The *consistency rate* is therefore computed as the complementary of the inconsistency rate as follows: $C = 1-I$. Fig. A5 reports the new agreement and (in-)consistency rates for all conditions based on Fig. A4, along with their magnitude. For example, participants expressed a better agreement for the Previous document ($AR(previous, \text{doc}) = 0.714$) than for Previous photo (0.393) and Previous video (0.357). In some cases, the media-specific agreement rate could maximal (*e.g.*, $AR(\textit{fullscreen on}, \text{photo})$=1 since 6 participants coincidentally proposed gestures of the same category, as depicted by a green rectangle on the third line of Fig. A4) or minimal (*e.g.*, $AR(\textit{fullscreen off}, \text{doc})$=0 since all 6 participants proposed a different gesture, as depicted by the red rectangle on the fourth line of Fig. A4). This shows the media-specific agreement rates in extreme conditions.

Overall, the best consistency rates are obtained for videos (74%), maps (69%), documents (68%), and photos (65%), respectively. For the Previous referent, the inter-media consistency rate is 78%, which is similar to the one obtained its symmetric referent Next among all media.

Appendix B. Description of the Gesture Recognizers

B.1. Static Gesture Recognizers

**$P^3+** consists of a 3D version of *$P+* algorithm (Vatavu, 2017). It is position-, direction-, and scale-invariant, but not rotation-invariant.

GPSDa is a novel position-, scale-, direction-, and rotation-invariant recognizer for hand poses regardless of the position of the hand. Its rotation-invariance makes it well suited to situations where hand poses should be recognized regardless of their direction (*e.g.*, rotate a picture by performing a GRAB gesture and gradually rotating the hand).

B.2. Dynamic Gesture Recognizers

3 cent is an optimized 3D enhancement of $1 (Wobbrock et al., 2007) by Caputo *et al.* Caputo et al. (2017), which recognizes uni-stroke gestures in two phases: the gesture is first normalized (similarly to $P) and its distance to each template is then computed as the sum of the squared distances of corresponding points. The template with the shortest distance is returned. 3¢ is position- and scale-invariant, but neither rotation- nor direction-invariant.

$P^3 generalizes $P (Vatavu, 2012) towards supporting 3D multi-stroke gestures which is similar to the $P3D recognizer implemented by (Cook et al., 2016). However, unlike $P3D, $P^3 does not support 3D static poses and 2D dynamic gestures recognition. To keep memory usage and execution time low, gestures are represented as unordered sets of points, called *point clouds*. Gesture recognition happens in two phases: normalization and cloud-matching. The normalization process is similar to other $-family algorithms and is divided into three steps: (1) resample the gesture to a fixed number of equidistant points; (2) scale the gesture uniformly to keep its shape; (3) translate the centroid of the point cloud to the origin, *i.e.*, $(0, 0, 0)$. The cloud-matching phase matches a gesture's point cloud to the point cloud of each template by associating each point from the template to exactly one point from the gesture. It then computes the resulting distance, as the sum of Euclidean distances between all pairs of matching points, and returns the template that minimizes it. This recognizer is position-, direction-, and scale-invariant, but not rotation-invariant. The modifications to the original $P algorithm are highlighted in blue. POINT is a structure with the `x`, `y`, `z`, and `strokeId` properties. POINTS is a list of POINT and TEMPLATES is a list of POINTS with their associated gesture class data (*e.g.*, the name of the gesture). Please note that the templates should be pre-processed to improve performance, but this is not shown in the pseudocode to keep it concise.

P^3RECOGNIZER(POINTS `pts`, TEMPLATES `templates`)

```
1:  n ← 32                                              ▷ Number of points
2:  NORMALIZE(PTS, N)
3:  score ← ∞
4:  for each template in templates do
5:      NORMALIZE(TEMPLATE, N)                          ▷ Should be pre-processed
6:      d ← GREEDYCLOUDMATCH(pts, template, n)
7:      if d < score then
8:          score ← d
9:          result ← template
10: return template
```

GreedyCloudMatch(Points ptsA, Points ptsB, int n)

```
1: e ← .50
2: step ← ⌊n^(1−e)⌋
3: min ← ∞
4: for i = 0 to n − 1 step step do
5:     d_1 ← CloudDistance(ptsA, ptsB, n, i)
6:     d_2 ← CloudDistance(ptsB, ptsA, n, i)
7:     min ← Min(min, d_1, d_2)
8: return min
```

CloudDistance(Points ptsA, Points ptsB, int n, int start)

```
1: matched ← new bool [n]
2: sum ← 0
3: i ← start
4: repeat
5:     min ← ∞
6:     for each j such that not matched[j] do
7:         d ← EuclideanDistance(ptsA[i], ptsB[j])
8:         if d < min then
9:             min ← d
10:            index ← j
11:    matched[index] ← true
12:    weight ← 1 − ((i − start + n)%n)/n
13:    sum ← sum + weight ∗ min
14:    i ← (i + 1)%n
15: until i == start
16: return sum
```

Resample(Points pts, int n)

```
1: I ← PathLength(pts)/(n − 1)
2: D ← 0
3: resampledPts ← {pts[0]}
4: for each p_i in pts such that i ≥ 1 do
5:     if p_i.strokeId == p_{i−1}.strokeId then
6:         d ← EuclideanDistance(p_i, p_{i−1})
7:         if (D + d) ≥ I then
8:             q.x ← p_{i−1}.x + ((I − D)/d) ∗ (p_i.x − p_{i−1}.x)
9:             q.y ← p_{i−1}.y + ((I − D)/d) ∗ (p_i.y − p_{i−1}.y)
10:            q.z ← p_{i−1}.z + ((I − D)/d) ∗ (p_i.z − p_{i−1}.z)
11:            Append(resampledPts, q)
12:            Insert(pts, i, q)                ▷ q will be the next p_i
13:            D ← 0
14:        else
15:            D ← D + d
16: while resampledPts.length ≤ n do
17:    Append(resampledPts, pts[pts.length - 1])
18: return resampledPts
```

NORMALIZE(POINTS pts, int n)

```
pts ← RESAMPLE(pts, n)
SCALE(pts)
TRANSLATETOORIGIN(pts, n)
```

PATHLENGTH(POINTS pts)

```
d ← 0
for each p_i in pts such that i ≥ 1 do
    if p_i.strokeId == p_{i−1}.strokeId then
        d ← d + EUCLIDEANDISTANCE(p_i, p_{i−1})
return d
```

SCALE(POINTS pts)

```
x_min ← ∞, x_max ← 0, y_min ← ∞, y_max ← 0, z_min ← ∞, z_max ← 0
for each p in pts do
    x_min ← MIN(x_min, p.x)
    y_min ← MIN(y_min, p.y)
    z_min ← MIN(z_min, p.z)
    x_max ← MAX(x_max, p.x)
    y_max ← MAX(y_max, p.y)
    z_max ← MAX(z_max, p.z)
scale ← MAX(x_max − x_min, y_max − y_min, z_max − z_min)
for each p in pts do
    p ← ((p.x − x_min)/scale, (p.y − y_min)/scale, (p.z − z_min)/scale, p.strokeId)
```

TRANSLATETOORIGIN(POINTS pts, int n)

```
c ← (0, 0, 0)
for each p in pts do                          ▷ Compute the centroid
    c ← (c.x + p.x, c.y + p.y, c.z + p.z)
c ← (c.x/n, c.y/n, c.z/n)
for each p in pts do                          ▷ Translate each point
    p ← (p.x − c.x, p.y − c.y, p.z − c.z, p.strokeId)
```

$P³+ is a more accurate version of $P³, adapted from $P+ (Vatavu, 2017). It brings three key improvements to $P³: (1) each point from the template can now be matched to more than one point of the gesture; (2) the distance between a gesture and a template takes into account the connections between consecutive points (in the form of their turning angles); (3) early abandonment of templates. This recognizer shares the same invariance properties as $P³. We only show the parts that were updated from $P³. The modifications to the original algorithm ($P+) are highlighted in blue.

P^3+Recognizer(Points pts, Templates templates)

```
n ← 32                                                    ▷ Number of points
Normalize(pts, n)
score ← ∞
for each template in templates do
    Normalize(template, n)                                ▷ Should be pre-processed
    d ← Min(CloudDistance(pts, template, n, score),
            CloudDistance(template, pts, n, score))
    if d < score then
        score ← d
        result ← template
return template
```

Q^3 is a 3D variant of $Q (Vatavu et al., 2018), which achieved a 46X speedup on average over *$P* with no loss of accuracy. It brings two key changes to P^3: (1) early abandonment of templates, as soon as the distance exceeds the current shortest distance; (2) each point cloud has a 16×16×16 3D look-up table (LUT), where each location (x, y, z) refers to the closest point. These LUTs allow Q^3 to compute a lower bound of the distance between a gesture and a template in $\mathcal{O}(n)$. If it exceeds the current shortest distance, the template can be rejected without computing the exact distance. It has the same invariance properties as P^3. We only show the parts that were updated from P^3. The modifications to the original algorithm ($Q) are highlighted in blue.

CloudDistance(Points ptsA, Points ptsB, int n, float minSoFar)

```
matched ← new bool [n]
sum ← 0
for i = 0 to n − 1 do          ▷ Match points from cloud ptsA with points from ptsB; one-to-many matchings allowed
    min ← ∞
    for j = 0 to n − 1 do
        d ← PointDistance(ptsA[i], ptsB[j])
        if d < min then
            min ← d
            index ← j
    matched[index] ← true
    sum ← sum + min
    if sum ≥ minSoFar then
        return sum
    for each j such that not matched[j] do   ▷ Match remaining points from cloud ptsB with points from ptsA; one-to-many matchings allowed
        min ← ∞
        for i = 0 to n − 1 do
            d ← PointDistance(ptsA[i], ptsB[j])
            if d < min then
                min ← d
        sum ← sum + min
        if sum ≥ minSoFar then
            return sum
        return sum
```

PointDistance(Point a, Point b)

1: **return** $\sqrt{(\texttt{a.x}-\texttt{b.x})^2+(\texttt{a.y}-\texttt{b.y})^2+(\texttt{a.z}-\texttt{b.z})^2+(\texttt{a.}\theta-\texttt{b.}\theta)^2}$

Normalize(Points pts, int n)

1: pts ← Resample(pts, n)
2: Scale(pts)
3: TranslateToOrigin(pts, n)
4: ComputeNormalizedTurningAngles(pts, n)

ComputeNormalizedTurningAngles(Points pts, int n)

1: $\texttt{pts}[0].\theta \leftarrow 0$
2: $\texttt{pts}[n].\theta \leftarrow 0$
3: **for** $\texttt{i} = 2$ **to** $\texttt{n} - 2$ **do**
4: $\quad \texttt{dpX} \leftarrow (\texttt{pts}[i+1].x - \texttt{pts}[i].x) * (\texttt{pts}[i].x - \texttt{pts}[i-1].x)$
5: $\quad \texttt{dpY} \leftarrow (\texttt{pts}[i+1].y - \texttt{pts}[i].y) * (\texttt{pts}[i].y - \texttt{pts}[i-1].y)$
6: $\quad \texttt{dpZ} \leftarrow (\texttt{pts}[i+1].z - \texttt{pts}[i].z) * (\texttt{pts}[i].z - \texttt{pts}[i-1].z)$
7: $\quad \texttt{pts}[i].\theta \leftarrow \frac{1}{\pi} \arccos\left(\frac{\texttt{dpX}+\texttt{dpY}+\texttt{dpZ}}{\|\texttt{pts}[i+1]-\texttt{pts}[i]\| * \|\texttt{pts}[i]-\texttt{pts}[i-1]\|}\right)$

$Q^3Recognizer(Points pts, Templates templates)

1: n ← 32 ▷ Number of points
2: m ← 16 ▷ Size of the LUT
3: Normalize(pts, n, m)
4: score ← ∞
5: **for each** template **in** templates **do**
6: $\quad$ Normalize(template, n, m) ▷ Should be pre-processed
7: $\quad$ d ← CloudMatch(pts, template, n, score)
8: $\quad$ **if** d < score **then**
9: $\quad\quad$ score ← d
10: $\quad\quad$ result ← template
11: **return** template

CloudMatch(Points pts, Points template, int n, int min)

1: e ← .50
2: step ← $\lfloor n^{1-\texttt{e}} \rfloor$
3: $\texttt{LB}_1$ ← ComputeLowerBound(pts, template, step, template.LUT)
4: $\texttt{LB}_2$ ← ComputeLowerBound(template, pts, step, pts.LUT)
5: min ← ∞
6: **for** $\texttt{i} = 0$ **to** $\texttt{n} - 1$ **step** step **do**
7: $\quad$ **if** $\texttt{LB}_1[\texttt{i/step}] < \texttt{min}$ **then**
8: $\quad\quad$ min ← Min(min, CloudDistance(pts, template, n, i, min))
9: $\quad$ **if** $\texttt{LB}_2[\texttt{i/step}] < \texttt{min}$ **then**
10: $\quad\quad$ min ← Min(min, CloudDistance(template, ps, n, i, min))
11: **return** min

CloudDistance(Points pts, Points template, int n, int start, float minSoFar)

```
1: unmatched ← {0, 1, 2, ..., n − 1}
2: i ← start
3: weight ← n
4: sum ← 0
5: repeat
6:     min ← ∞
7:     for each j in unmatched do
8:         d ← SqrEuclideanDistance(pts[i], template[j])
9:         if d < min then
10:            min ← d
11:            index ← j
12:    Remove(unmatched, index)
13:    sum ← sum + weight ∗ min
14:    if sum ≥ minSoFar then
15:        return sum
16:    weight ← weight − 1
17:    i ← (i + 1)%n
18: until i == start
19: return sum
```

ComputeLowerBound(Points pts, Points template, int step, int[] LUT)

```
1: LB ← new float[n/step + 1]          ▷ Multiple lower bounds, one for each starting point
2: SAT ← new float[n]                  ▷ Summed area table for fast computations
3: LB[0] ← 0
4: for i = 0 to n − 1 do               ▷ First, compute the lower bound for starting point index 0
5:     index ← LUT[pts[i].x, pts[i].y, pts[i].z]
6:     d ← SqrEuclideanDistance(pts[i], template[index])
7:     if i == 0 then
8:         SAT[i] ← d
9:     else
10:        SAT[i] ← SAT[i − 1] + d
11:    LB[0] ← LB[0] + (n − i) ∗ d
12: for i ← step to n − 1 step step do   ▷ Compute the lower bound for the other starting points
13:    LB[i/step] ← LB[0] + i ∗ SAT[n − 1] − n ∗ SAT[i − 1]
14: return LB
```

Normalize(Points pts, int n, int m)

```
1: pts ← Resample(pts, n)
2: Scale(pts)
3: TranslateToOrigin(pts, n)
4: LUT ← ComputeLUT(m, pts, n)
```

```
Scale(Points pts, int m)
x_min ← ∞, x_max ← −∞, y_min ← ∞, y_max ← −∞, z_min ← ∞, z_max ← −∞
for each p in pts do
    x_min ← Min(x_min, p.x)
    y_min ← Min(y_min, p.y)
    z_min ← Min(z_min, p.z)
    x_max ← Max(x_max, p.x)
    y_max ← Max(y_max, p.y)
    z_max ← Max(z_max, p.z)
scale ← Max(x_max − x_min, y_max − y_min, z_max − z_min)/(m − 1)
for each p in pts do
    p ← ((p.x − x_min)/scale, (p.y − y_min)/scale, (p.z − z_min)/scale, p.strokeId)
```

```
ComputeLUT(Points pts, int n, int m)
LUT ← new int[m, m, m]
for x = 0 to m − 1 do
    for y = 0 to m − 1 do
        for z = 0 to m − 1 do
            min ← ∞
            for i = 0 to n − 1 do
                d ← SqrEuclideanDistance(pts[i], newPoint(x, y, z))
                if d < min then
                    min ← d
                    index ← i
            LUT[x, y, z] ← index
return LUT
```

```
SqrEuclideanDistance(Point a, Point b)
return (a.x − b.x)^2 + (a.y − b.y)^2 + (a.z − b.z)^2
```

Jackknife is a general-purpose 3D gesture recognizer (Taranta et al., 2017) that supports any number of articulations. Unlike most $-family recognizers, it represents gestures as time-series. It uses the nearest neighbor approach, where it compares a gesture to each template and returns the closest one. It uses *Dynamic Time Warping* as a distance measure but applies a series of correction factors to take into account differences in gesture scale and span. As a result, this recognizer is both position- and scale-invariant, but neither direction- nor rotation-invariant.

Appendix C. Initial Gesture Elicitation Studies for Photo- and Video-Browsing

Two experiments were conducted to determine users' preferred gestures for browsing photos and videos. 30 participants (16 female) aged between 18 and 77 years old (M=34.5, SD=17.5) took part in the photo-browsing experiment. 22 participants (8 females) aged between 15 and 54 years old (M=28.9, SD=12.6) took part in the video-browsing experiment. Agreement rates ranged between .064 and .690 (M=.205, SD=.147) for the photo-browsing experiment (Table C1) and between .052 and .593 (M=.229, SD=.162) for the video-browsing experiment (Table C2).

Referent	AR (magnitude)	First most agreed gesture	Second most agreed gesture
1. *Pan Photo Gallery Right*	.251 (M)	1 hand flick	Multiple fingers swipe
2. *Pan Photo Gallery Left*	.269 (M)	Multiple fingers swipe	1 finger swipe
3. *Pan Photo Gallery Up*	.170 (M)	1 finger swipe	Hand rotation
4. *Pan Photo Gallery Down*	.175 (M)	1 hand flick	1 finger swipe
5. *Go to Next Page of Photo Gallery*	.205 (M)	Multiple fingers swipe	1 hand flick
6. *Go to Previous Page of Photo Gallery*	.168 (M)	Multiple fingers swipe	1 hand flick
7. *Zoom in a Photo*	.338 (H)	Splay the hand	Bring hands further apart
8. *Zoom out a Photo*	.299 (M)	Clench the hand	Bring hands further apart
9. *Maximize a Photo from Normal View to Full Screen*	.363 (H)	Bring hands further apart	Splay the hand
10. *Unmaximize a Photo from Full Screen to Normal View*	.193 (M)	Clench the hand	Bring hands closer
11. *Open Photo Information*	.308 (H)	1 finger tap	Splay the hand
12. *Close Photo Information*	.170 (M)	1 finger tap	1 finger swipe
13. *Set Layout to Carousel*	.099 (L)	Hand rotation	1 finger tap
14. *Start Slideshow*	.126 (M)	1 finger tap	Hand pose
15. *Stop Slideshow*	.156 (M)	1 finger tap	Move flat hand forward/backward
16. *Take a Selfie*	.099 (L)	Hand pose	1 finger tap
17. *Insert a new Photo*	.092 (L)	Hand pose	1 finger tap
18. *Duplicate an Existing Photo*	.097 (L)	1 finger swipe	1 finger tap
19. *Delete an Existing Photo*	.094 (L)	1 finger swipe	Hand pose
20. *Crop a Photo*	.090 (L)	Draw	Bring hands further apart
21. *Resize a Photo*	.333 (H)	Bring hands further apart	Splay the hand
22. *Rotate a Photo 90° Clockwise*	.634 (V)	Hand rotation	N/A
23. *Rotate a Photo 90° Counter-clockwise*	.690 (V)	Hand rotation	N/A
24. *Increase Constrast*	.071 (L)	1 finger swipe	Open arms (with flat hands)
25. *Decrease Contrast*	.064 (L)	1 finger swipe	Multiple fingers swipe
26. *Dock a Photo*	.097 (L)	1 finger swipe	1 hand flick
27. *Undock a Photo*	.152 (M)	1 finger swipe	Take and throw
28. *Like a Photo*	.218 (M)	1 finger tap	Hand pose
29. *Search by Criteria*	.131 (M)	Hand pose	Draw
30. *Share a Photo with Somebody (vocal email)*	.087 (L)	Hand pose	1 finger swipe
31. *Convert a Photo into Another Format*	.115 (M)	Bring hands further apart	1 finger tap

Table C1.: First and second most agreed gesture proposals for each referent (photo-browsing experiment). Referents with above average agreement rates are highlighted in blue. Notation for the Agreement Rate (AR): L=low, M=medium, H=high, V=very high (Vatavu & Wobbrock, 2015).

Referent	AR (magnitude)	First most agreed gesture	Second most agreed gesture
1. *Pan Video Gallery Right*	.433 (H)	1 hand drag	1 finger drag
2. *Pan Video Gallery Left*	.433 (H)	1 hand drag	1 finger drag
3. *Pan Video Gallery Up*	.390 (H)	1 hand drag	1 finger drag
4. *Pan Video Gallery Down*	.390 (H)	1 hand drag	1 finger drag
5. *Go to Next Page of Video Gallery*	.407 (H)	1 hand drag	1 finger drag
6. *Go to Previous Page of Video Gallery*	.329 (H)	1 hand drag	1 finger drag
7. *Zoom in a Video*	.182 (M)	1 hand drag	Close one hand
8. *Zoom out a Video*	.242 (M)	1 hand drag	Close one hand
9. *Set a Video in Full Screen*	.165 (M)	1 finger tap	Drag 1 finger from each hand
10. *Unmaximize a Video from Full Screen to Normal View*	.095 (L)	Drag 1 finger from each hand	2 hands drag
11. *Play a Video*	.333 (H)	1 finger tap	1 finger drag
12. *Stop Playing a Video*	.294 (M)	Push/pull 1 hand	1 finger tap
13. *Move to the Beginning of Video*	.169 (M)	1 hand drag	1 finger tap
14. *Move to the End of a Video*	.169 (M)	1 hand drag	1 finger tap
15. *Open Video Information*	.117 (M)	1 finger tap	Push/pull 1 hand
16. *Close Video Information*	.117 (M)	1 hand drag	1 finger drag
17. *Set Layout to Carousel*	.117 (M)	1 hand drag	Tap and drag 1 finger
18. *Insert a New Video*	.078 (L)	Push/pull and drag 1 hand	Push/pull 1 hand
19. *Duplicate a Video*	.052 (L)	2 hands drag	Tap and drag 1 finger from each hand
20. *Delete a Video*	.074 (L)	1 hand drag	1 finger drag
21. *Rotate a Video 90° Clockwise*	.593 (V)	1 hand drag	Push/pull 1 hand
22. *Rotate a Video 90° Counter-clockwise*	.593 (V)	1 hand drag	Push/pull 1 hand
23. *Dock a Video*	.078 (L)	Push/pull and drag 1 hand	Tap and drag 1 finger
24. *Undock a Video*	.078 (L)	Push/pull and drag 1 hand	Close, push/pull and drag 1 hand
25. *Like a Video*	.091 (L)	1 finger drag	Push/pull 1 hand
26. *Search for a Video by Criteria*	.082 (L)	1 finger drag	1 hand drag
27. *Share a Video with Somebody*	.082 (L)	1 hand drag	Close, push/pull and drag 1 hand

Table C2.: First and second most agreed gesture proposals for each referent (video-browsing experiment). Referents with above average agreement rates are highlighted in blue. Notation for the Agreement Rate (AR): L=low, M=medium, H=high, V=very high (Vatavu & Wobbrock, 2015).