Understanding Shortcut Gestures on Mobile Touch Devices

Benjamin Poppinga[1], Alireza Sahami Shirazi[2], Niels Henze[2], Wilko Heuten[3], Susanne Boll[1]

[1]University of Oldenburg
Oldenburg, Germany
firstname.lastname@uni-oldenburg.de

[2]VIS, University of Stuttgart
Stuttgart, Germany
firstname.lastname@vis.uni-stuttgart.de

[3]OFFIS - Institute for Information Technology
Oldenburg, Germany
heuten@offis.de

ABSTRACT

Touch gestures become steadily more important with the ongoing success of touch screen devices. Compared to traditional user interfaces, gestures have the potential to lower cognitive load and the need for visual attention. However, nowadays gestures are defined by designers and developers and it is questionable if these meet all user requirements. In this paper, we present two exploratory studies that investigate how users would use unistroke touch gestures for shortcut access to a mobile phone's key functionalities. We study the functions that users want to access, the preferred activators for gesture execution, and the shapes of the user-invented gestures. We found that most gestures trigger applications, letter-shaped gestures are preferred, and the gestures should be accessible from the lock screen, the wallpaper, and the notification bar. We conclude with a coherent, unambiguous set of gestures for the 20 most frequently accessed functions, which can inform the design of future gesture-controlled applications.

Author Keywords

Touch gestures; shortcut; mobile phone; gesture recognition; instant access; quick launch.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Input devices and strategies*

INTRODUCTION

Over the past years, mobile devices, such as smart phones, tablets or wearable devices, started to replace traditional desktop computers as the most successful computing devices. Many of these mobile devices use touch-sensitive displays or surfaces as primary means of interaction, often mimicking traditional user interfaces like keyboards. However, with smaller form factors and display sizes, touch-based interaction runs into limitations as buttons and other interactive elements become too small to be operated accurately. Thus, alternative interaction techniques have been invented and studied. Most observable, we saw a breakthrough of gesture-based interaction on many recent mobile devices.

On nowadays devices a small set of simple touch gestures is used for zooming, rotating, and scrolling. Further, touch gestures are also used to provide users with a *shortcut* to certain functionalities, e.g., a swipe gesture to unlock the phone or to show the device's camera. Typically, these gestures were defined by the designers and developers of a system [13], although previous work showed that gesture sets that are mainly informed by the designers' intuition might not satisfy the users' requirements [23]. Beyond a basic set of well-established gestures, touch gestures for mobile phones are still not completely understood. For designers and developers it is unclear which action a gesture should cause, how the gesture should be shaped, and where the gestures should be executed.

In this paper, we study unistroke touch gestures for instant access to a mobile phones' core functions in the wild and from a user's perspective. To do this we design our apparatus, Gestify, which allows users to train and recognize custom gestures on Android phones. We deploy this application on the Google Play store and observe how users create and use gestures in their real life. Further, we conduct a supplemental lab study with 18 participants to substantiate these observations and to collect additional qualitative insights. We study which actions users prefer to cause with their gestures, which shapes their gestures have, and how users trigger gestures in their everyday life.

The contributions of this paper are as follows:

- We show that gestures are a valuable means for instant access and they should be executable on phones' lock screens, notification bars, and wallpapers.
- We highlight the 20 most important functions that should be accessible through shortcut gestures.
- We identify that *letter*-shaped gestures, which represent the first character of a related action, are used most commonly.
- We provide a consistent gesture set for the 20 most important functions.

MobileHCI 2014, September 23–26, 2014, Toronto, ON, Canada.

ACM 978-1-4503-3004-6/14/09…$15.00.
http://dx.doi.org/10.1145/2628363.2628378

In the next section, we provide an overview of previous work on touch and shortcut gestures on mobile devices. Afterwards, we describe the apparatus that we used to conduct two users studies. We describe an 'in the wild' field study to observe users' real life behavior, and a controlled lab study to substantiate the findings. We discuss the results of the studies and derive a consistent gesture set for the 20 most important functions. Finally, we conclude and provide potential directions for future work.

RELATED WORK

Gestures have a long history in human-computer interaction (HCI). They are one facet of interaction techniques that are sometimes dubbed natural user interfaces [22], and their exists a large body of work which investigated a broad range of different gestural interfaces (see [4] for an overview).

Previous work mainly investigated three different types of gestural interfaces: free-hand gestures in mid-air, gestures as alternatives for traditional user interface elements, and touch gestures. Work on free-hand gestures that are executed in mid-air most closely resemble gestures that humans use for communicating with each other [5]. With the release of the Wii and the Kinect, free-hand gestures became widely used for gaming and different applications in HCI research [19]. In desktop computing, gestures were mostly investigated as alternatives or complements for direct manipulation interfaces. Moyle and Cockburn, for example, analyzed mouse and pen flick gestures [14]. With the recent success of touch screen devices, touch gestures became the most widely used type of gestural interface and a focus in HCI research.

Touch gestures are meaningful strokes which are executed on a touch-sensitive surface, like a mobile phone, a tablet or a larger tabletop computer (see [26] for a review). It has been shown that touch gestures have a similar performance as keyboard shortcuts, but come with significant cognitive advantages, i.e., they are easier to learn and recall [3]. In addition, stroke gestures can be used eyes-free and are particularly beneficial for selecting frequently executed commands [17]. The variety of different application domains, settings, and techniques is huge, e.g., [8, 6, 12, 20]. The most of these domains and applications are investigated on a by-case basis. Consequently, a practical, coherent understanding of touch gestures is hardly existing and no comprehensive guidelines for gestural control are available [25].

Previous work not only investigated the design of touch gestures but also explored design processes for gestural interfaces. In general, it has been found that user-defined gestures are easier to memorize than pre-designed gestures [15]. Actual designs for user-defined gestures are typically identified in participatory design studies [23, 10, 9, 20]. Most prominently, Wobbrock et al. [23] presented an approach to design tabletop gestures with the help of potential users. They showed the actual effect a gesture would cause and then ask the user to define an appropriate action, i.e., gesture. In a lab study with 20 participants they recorded gestures, performed with one or two hands. They found that users prefer to execute gestures with one hand and mostly do not care about how many fingers are used. Wobbrock et al. conclude with a user-defined standard gesture set for common surface computer tasks and a taxonomy of surface gestures. Seyed et al. applied a similar approach for multi-display environments [20].

Ouyang and Li explored user-defined gesture shortcuts on mobile phones to realize a crowdsourced gesture recognition system [16]. They explored user-defined gesture shortcuts of 26 Android users and found that about 72 % of all gestures were of alphanumeric nature, i.e., a character or letter. They observed that users often define similar gestures for different actions, resulting in ambiguity and a decrease of recognition rate to about 70 %. Further, they reported that only a few gesture actions were shared by all users. Thus, in most cases the users defined gestures for very individual actions.

Another interest for the community lies in the application and everyday use of gestures [11, 21]. Gesture Search by Li [11] allows users to search their mobile phone for apps, contacts, and media files with the help of letter-shaped stroke gestures. Thereby, a variety of actions can be controlled, e.g., application launches, phone book access, or music playback. The research focuses on how the gesture recognition can be combined with traditional user interface elements. They concluded that Gesture Search enables a quick and easy way to access mobile data and users appreciate the usefulness and usability.

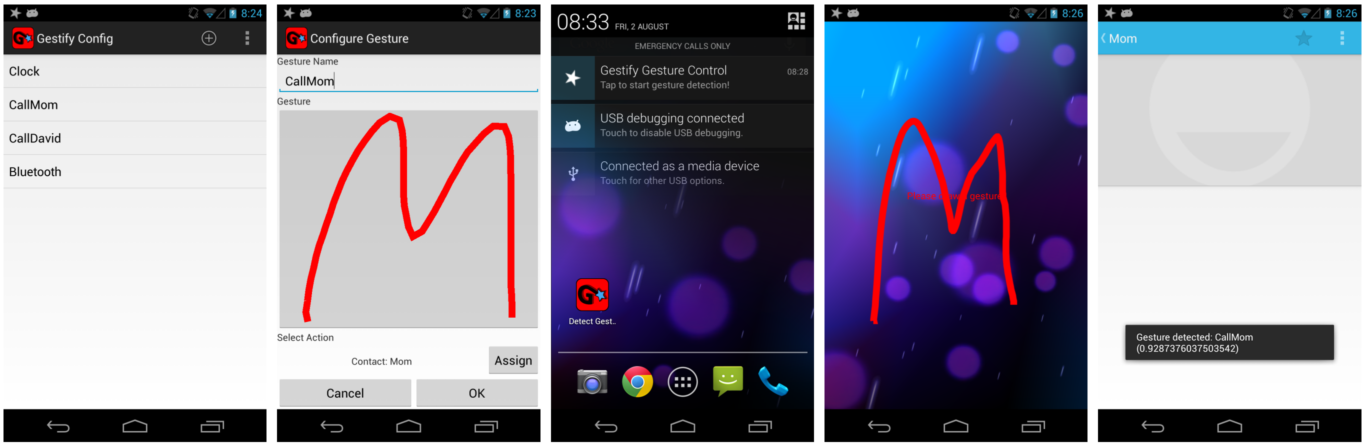
In summary, most of existing work provides valuable insights into a single research question, e.g., what do users want to control via gestures, how should a gesture look like, and how can gestures be applied in various settings or scenarios. Few papers study gestures in context or in the wild, and come with other limitations. For example, Gesture Search [11] was limited to letter-shaped gestures and their quick start application wasn't accessible from anywhere in the operating system. In contrast, we follow an exploratory approach and investigate the overall nature of touch gestures for shortcut access on mobile phones. With this approach we derive a coherent gesture set, which combines our findings on the gesture actions and ideal gesture shapes, that should be accessible through a phone's lock screen, notification bar, and wallpaper.

APPARATUS: GESTIFY

To investigate how touch gestures as shortcut to the phone's functionalities are used, we developed the Gestify application, that enables users to define gestures and use them to access various functions of their phone. In the following, we describe the design of the application, the used gesture recognition algorithm and the data that is recorded.

Creating and Managing Gestures

Gestify is an Android application, running on Android 3.0 or newer, that consists of two sub applications: *Gesture Manager* and *Gesture Recognizer*. With installation of the application both sub applications are installed and added to the home screen. The Gesture Manager allows users to manage existing gestures and define new ones, the Gesture Recognizer does the actual gesture recognition.



(a) List of existing gestures.

(b) Add a new gesture.

(c) Trigger detection through one of the provided activators, e.g., notification bar.

(d) Draw gesture in recognition activity.

(e) A pop-up indicates that the gesture is detected.

Figure 1. Gestify is an Android application that allows users to define and execute their own shortcut gestures. If gestures were trained to the system they can be recognized via different activators, i.e., through the lock screen, the notification bar, the wallpaper, and through a separate activity. If a gesture is detected, a toast is shown and the linked action is executed.

When the Gesture Manager is launched, an initially empty list shows the names of all existing gestures (see Figure 1a). A new gesture can be created by pressing the add button in the top right menu. The subsequently appearing dialogue is shown in Figure 1b and first asks the user to name the new gesture. The name can consist of alphanumerical characters and is recorded solely to allow users to maintain an overview over existing gestures (as shown in Figure 1a).

Gestify only supports unistroke gestures as defined in [26]. Thus, the touch movement for defining a gesture has to be a single, continuous movement that can have a theoretically unlimited complexity and length. Consequently, a single tap or two successive strokes are no valid gestures. Although there are multistroke gesture recognition engines available, e.g., [2], we decided for a unistroke recognition to emphasize the shortcut characteristic of the gestures.

In the last step of the creation process an action needs to be defined, which is triggered once the gesture is recognized. The user can select to start any application that is installed on his or her phone (*application gestures*), show a contact from the phone book (*contact gestures*) or change a system setting (*setting gesture*). While any application or contact can be selected, only core system settings, i.e., Bluetooth, Wi-Fi, Ringer, Vibration, can be changed for technical reasons. The user can decide to enable, disable or toggle one of these system settings. A tap on the *OK* button initiates the actual gesture training, closes the dialogue, and guides the user back to the gesture overview.

To simplify the analysis of recorded gestures, once created, a gesture cannot be updated. After a gesture has been designed and associated with an action, neither the name, the shape nor the action can be modified. If the user wants to change a gesture, it must be deleted and a new one has to be created.

Using Shortcut Gestures

The second sub application, Gesture Recognizer, is the interface for the actual gesture recognition. The app includes a single, full screen view in which the user can draw a gesture. A drawn gesture is taken as input for the gesture recognition engine and, based on the gesture recognition result, the corresponding action assigned to the gesture is executed. The view can be reached through four different ways, namely, (1) the lock screen, (2) the phone's notification bar, (3) directly launching the Gesture Recognizer app, and (4) a live wallpaper. In the following, we refer to these different ways to trigger the gesture recognition view as *activators*.

By installing Gestify and respectively the Gesture Recognizer app, the gesture recognition view automatically replaces the device's current lock screen. Further, this view can be reached through a short cut in the Android notification bar (see Figure 1c). The lock screen and notification bar activators are enabled by default, although they can be individually disabled through the application settings. In addition, the gesture recognition view can serve as a live wallpaper, which allows to recognize gestures straight from the phone's home screen. Due to technical restrictions users need to manually enable the wallpaper activator. To avoid confusion with other wallpaper-related interactions, e.g., changing the home screen with a sideways swipe, the gesture recognition needs to be initiated through a double tap onto an empty area. A pop-up briefs the user about this procedure.

Visual feedback is provided while drawing a gesture (see Figure 1d) to increase the drawing accuracy [1]. The actual recognition is triggered once the finger is lifted from the screen. The result for each recognition trial is displayed as a pop-up, which either shows the gesture label and recognition probability or that no gesture could be detected (see

Figure 1e). If a gesture is recognized, the according action is immediately executed.

Gesture Recognition Engine

Because the user should not be asked to provide several training gestures of the same type, we decided for a gesture recognition engine, which achieves high accuracies for few sample sizes. Gestify uses the $1 recognizer by Wobbrock et al. [24], which is based on geometric template matching, and achieves recognition rates of over 97 % with a single training instance. Although we did some empirical testing on the individual parameters of the engine, we eventually went with the best-performing default configuration [24]. That means every gesture is scaled to a 500x500 pixel square and consists of 64 points. The gesture recognition engine has some limitations, such as the inability to deal with ambiguous gestures. These limitations were not communicated to the users to avoid limiting them in their creativity.

Data Logging

The Gestify application is accompanied by a custom logging framework. Once the application is installed, various device information are recorded, e.g., the manufacturer, the model and the locale. Further, the logging framework records each created gesture, including their label and action, and when a gesture is deleted. All recognition attempts are stored with their origin, i.e., wallpaper, lock screen, notification bar, or the activity itself, and their recognition result. For ethical reasons no personal information, e.g., the contact name for gestures which trigger a phone call, is logged. The log data is transferred to a central server every two minutes.

'IN THE WILD' FIELD STUDY

In our first study, we investigate how people use the Gestify app 'in the wild', i.e., in their everyday life. We determine the actions that are triggered, the gesture shapes that are drawn, and the activators which are used to do the gesture recognition.

Recruitment

We published Gestify in the Google Play store in April 2013. Until January 2014 the application was downloaded and used by 388 users. The majority, 259 users (66.75 %) had an English locale, e.g., en_US, en_GB followed by German locales (de_DE) with 24 users (6.19 %). In total, 124 users (31.96 %) had an European/African time zone, i.e., GMT/0 to GMT/2 and 70 (18.04 %) an American timezone, i.e., GMT/-4 to GMT/-8. This shows that our application was mostly used by English- or German-speaking EU citizens and US Americans, which has to be considered when interpreting the results [7]. All users were using phones with Android 4.0 or newer.

Dataset

Gestify recorded information about the device, trained gestures and gesture recognition attempts. This data was transferred over the Internet and stored on a server. It was analyzed using the R software environment for statistical computing.

Our analysis shows that the 388 users trained 1134 gestures, whereby the number of trained gestures per user ranges from 1 to a maximum of 25 gestures. On average, each user trained 2.92 gestures (SD 3.61, median 1) and used the application for 5.15 days (SD 22.10, median 0.002). Given these numbers it is clear that the majority of users just tested the application for a short time and then stopped using it.

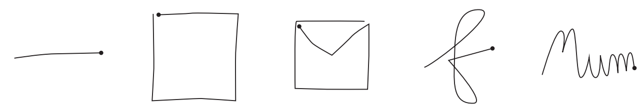


Figure 2. We empirically identified five different shapes each gesture can be assigned to. This figure shows an example for each of the shapes.

Application Categorization

In total, 303 unique applications were launched using Gestify. To investigate for which application category the gesture action was used, we retrieved application categories from the Google Play marketplace. However, as also reported by Sahami et al. [18], we realized that the categories are too generic to differentiate between application functionalities. Further, also the class/package names are inconsistent, rarely self-explaining and, thus, don't allow for a classification. Consequently, we manually investigated the underlying functionality of each application and assigned it to a unique and illustrative category. Thereby, we followed the approach by Sahami et al., but successively added more categories, when we realized this was necessary. This resulted in 17 categories. Categories had between 4 and 155 unique applications (mean 17.82, SD 35.80, median 7).

Gesture Shape Categorization

Gestify allows a user to use any type of gesture shape. For a better characterization and discussion of the various shapes, we manually analyzed and classified all gestures in a two-step process. In the first step, we went through all gestures and identified a set of five shape classes: *abstract*, *geometric*, *icon*, *letter* and *word* shapes (see Figure 2). *Abstract* gestures consists of not inherently meaningful, not closed movements, e.g., a straight swipe or a wave-like movement. *Geometric* gestures look like well-known, closed geometric shapes, e.g., a circle, a square, or a heart. *Icon* gestures recap parts of an app-icon or other well-known icons, e.g., an envelope (for a messenger application) or a pentagram. *Letter* gestures look like an alphabetic character or digit, and *word* gestures contain more than one letter or whole words. In the second step, we went through all gestures again and assigned each of them to a single category. In the majority of all cases the classification was obvious. In borderline cases we assigned a gesture to the more generic class, e.g., an *f*-shaped gesture for the Facebook application was assigned to the *letter* category and not to the *icon* category.

'Power Users' Subsample

To obtain a better understanding about how gestures are used, we extracted a sub-sample of users who used the application for a longer period. This allows us to study if the gesture actions, shapes, or activators differ between short-term and long-term users. We call these users *power users* and include

n	All Users		Power Users		n
471	Other	41.53 %	Other	44.66 %	113
111	Messenger	11.50 %	Messenger	13.76 %	30
82	Phone	8.50 %	Phone	10.55 %	23
75	Browser	7.78 %	Social Netw.	6.42 %	14
71	Camera	7.36 %	Music Player	5.96 %	13
58	Music Player	6.01 %	Camera	5.96 %	13
54	Gallery	5.60 %	Mail	5.50 %	12
52	Social Netw.	5.39 %	Gallery	4.59 %	10
41	Mail	4.25 %	Browser	3.21 %	7
27	Settings	2.80 %	Settings	2.29 %	5

Table 1. The top ten categories of application gestures. Most gestures were defined for messaging apps, such as WhatsApp or SMS.

only users who actively used the application for at least a full week, i.e., 7 days, and trained and used at least 1 gesture. This resulted in a set of 36 users, which closely resemble the original sample characteristics, i.e., locale and time zone. These power users trained a total number of 253 gestures. Each power users trained on average 7.03 gestures (SD 5.13, median 5.5) and used the application for 52.72 days (SD 53.18, median 29.79).

Results

In the following, we present the user-defined *actions* and visual *shapes* of gestures. Further, we study which *activator* users preferred to use and draw their gestures on. We do this analysis for both, all users and the identified 'power users' subsample.

Gesture Actions

Gestify differentiates between three types of possible gestures: start an application, show contact details, or modify a system setting. Of the 1134 recorded gestures, 965 gestures (85.10 %) were *application gestures*. 87 gestures (7.67 %) were *contact gestures*, and 82 gestures (7.23 %) were *setting gestures*. Power users contributed 253 gestures, of which 218 (86.16) are *application*, 19 (7.51 %) *settings*, and 16 (6.32 %) *contact gestures*.

An analysis of the application categories shows that messaging applications, such as WhatsApp or SMS/MMS messengers, were the most frequently launched applications (111 gestures, 11.50 %). Phone-related apps, such as the phone book or the dialer activity were the second most frequent applications (82 gestures, 8.50 %) followed by browser applications (the Android default browser, Chrome or Firefox) with 75 gestures (7.78 %) of the total 965 application gestures. For power users a similar trend was observed, except that the third-most often triggered functionality were social networks instead of browsers (14 of 218 gestures, 6.42 %). About one third of all application gestures (302 of 965, 31.30 % and 78 of 218, 35.78 % for power users) were not of a particular popular functionality, and therefore assigned to the 'other' category. For example, these gestures were used to trigger specific games or niche applications, which are of importance for the individual user, but not of general relevance. An overview of the top ten actions for both user groups can be found in Table 1.

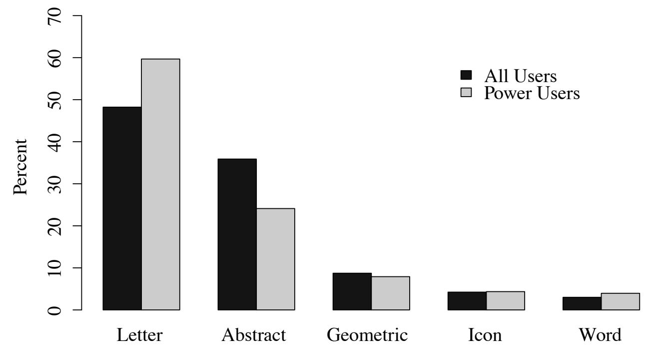


Figure 3. Gesture shapes for all users and power users. Letter-shaped gestures are the most prominent among both user groups, observed in about half of all cases.

From 82 *setting gestures* observed 9 were identified as erroneous and therefore were excluded from analysis. Of the remaining 73 gestures, 17 (23.29 %) were configured to toggle Wi-Fi, 14 (19.18 %) toggled Bluetooth, and 8 (10.96 %) enabled vibration. On average, each user uses a single gesture to toggle a system settings, i.e., turn on or off a setting depending on its current state. About 89.04 % of all gestures were configured to control Wi-Fi, Bluetooth, or vibration (about 30 % each). Only 10.96 % were used to change ringer settings.

We made slightly different observations for power users. They contribute 19 *setting gestures* in total, of which none was invalid. Of these, 6 gestures (31.58 %) were configured to enable vibration. 5 gestures (26.32 %) toggled Wi-Fi and 3 gestures (15.79 %) disabled vibration. 2 gestures each (10.53 %) were used to toggle Bluetooth or vibration. A single gesture (5.26 %) was set up to toggle the ringer. Similar to the set of all users, the majority of gestures (94.74 %) operates Wi-Fi, Bluetooth, or the vibration actuator.

Further, we observed 87 *contact gestures* for all users and 16 *contact gestures* for power users. However, for ethical reasons we did not record the meaning of any contact gestures, as this would allow us to track intimate contact details.

Gesture Shapes

Of the 1134 recorded gestures, *letter*-shaped gestures were the most frequent ones with 546 gestures (48.15 %) (see Figure 3). 407 gestures (35.89 %) matched the *abstract* shape, consisting of mostly simple swipes. The *geometric* shape included 99 gestures (8.73 %), the *icon* shape 48 gestures (4.23 %) and the *word* shape 34 gestures (3.00 %). We observed a similar distribution for power users, although the power users tend to have less *abstract* and more *letter*-shaped gestures.

For a more in-depth analysis of the gesture shapes, we extracted a set of gestures for the most commonly actions used. The set consists of the top 3 applications from the largest category, i.e., other applications (Play Store, YouTube, Google Maps), the top 2 applications from the second largest category, i.e., messenger applications (WhatsApp, SMS Composer), and the first application of every other category with

		Facebook (Social Network)	Phonebook (Phone)	Play Store (Others)	Camera (Camera)	WhatsApp (Messaging)	Gallery (Gallery)	Browser (Browser)	Settings (Settings)	SMS (Messaging)	Youtube (Other)	Mail (Mail)	Clock (Clock)	Calculator (Calculator)	Google Maps (Other)	Music (Music Player)	Calendar (Organiser)	Launcher (Launcher)	Flashlight (Flashlight)	Fav. Contact	Toggle Wifi
Field Study	n	39	37	32	29	27	26	25	24	16	15	13	13	13	13	13	11	5	2	87	17
Field Study	1st	31%	14%	28%	34%	81%	23%	28%	29%	38%	60%	38%	31%	23%	31%	31%	36%	20%	50%	9%	71%
Field Study	2nd	18%	11%	9%	21%	4%	12%	12%	13%	19%	13%	31%	15%	15%	15%	23%	18%	20%	50%	8%	12%
Lab Study	1st	59%	15%	39%	24%	61%	33%	21%	29%	28%	71%	28%	28%	47%	44%	41%	28%	28%	28%	30%	22%
Lab Study	2nd	18%	15%	22%	18%	6%	28%	16%	24%	22%	12%	17%	22%	18%	17%	18%	22%	17%	17%	24%	11%
Lab Study	n	17	18	18	17	18	18	18	17	18	17	18	18	17	18	17	18	18	18	17	18

Table 2. A visualization of the top two gestures for the twenty most popular gesture actions. For some actions the gestures are very consistent among users and studies, e.g., with about 71 % in the field study and 61 % in the lab study most people agree that *W*hatsApp should be triggered with a *W* gesture.

more than 10 trained gestures. Further, we included the most prominent *settings gesture* (i.e., Toggle Wi-Fi), and all *contact gestures*. The resulting set consists of 20 representative actions for the most commonly controlled functionalities and is shown in Table 2.

Regarding the shape of *application gestures*, we found that 459 of the 965 gestures (47.56 %) were classified to have a *letter* shape. Further, we identified 344 (35.65 %) *abstract*, 85 (8.81 %) *geometric*, 47 (4.87 %) *icon*, and 30 (3.11 %) *word* gestures. The in-depth analysis shows that there is often a strong consensus among users on which gesture shape should be used for which application (see Table 2). In fact, users often select the first letter of the application name, e.g., *W* to start *W*hatsApp or *f* to start *f*acebook. For power users we observed the same order with a similar distribution, i.e., 56.77 % were *letter* gestures, 24.77 % *abstract*, 8.72 % *geometric*, 5.05 % *icon*, and 4.59 % *word* gestures.

Again, the 9 erroneous *setting gestures* were excluded from the analysis. Of the remaining 73 *setting gestures*, 35 (47.95 %) were of *letter*, 25 (34.25 %) of *abstract*, 9 (12.33 %) of *geometric*, and 4 (5.48 %) of *word* shape. None of the gestures had an *icon* shape. *Setting gestures* of power users were 78.95 % *letter*-shaped, 15.79 % *abstract*-shaped, and 5.26 % of *geometric* shape. The two most common actions for all users, i.e., toggle Wi-Fi and toggle Bluetooth, were most frequently triggered with a *w*/*W*-shaped (12 of 17, 70.59 %) respectively *B*-shaped (7 of 14, 50.00 %) *letter* gesture (see Table 2).

We observed that 50 of 87 (57.47 %) *contact gestures* were *letter* shaped. 32 gestures (36.78 %) had an *abstract*, 4 (4.60 %) a *geometric*, and 1 (1.15 %) an *icon* shape. No *word*-shaped gestures were observed. Power users created 12 (75.00 %) *letter*-shaped *contact gestures*, and 4 (25.00 %) *abstract* gestures.

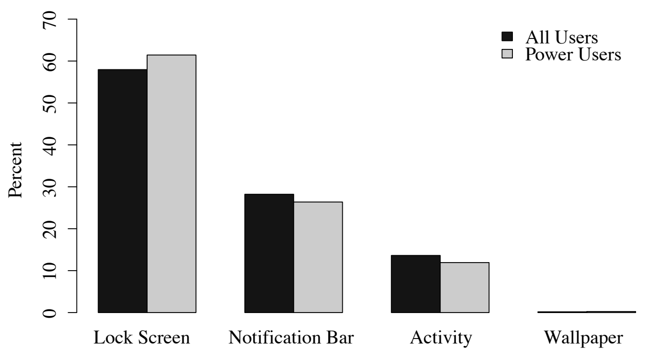


Figure 4. Users most frequently used the lock screen as activator for the gestures recognition. The wallpaper activator was hardly used at all, although study participants rated it best in the lab study.

Gesture Activators

Gestify offers four activators to start the gesture recognition. A user could use the lock screen, the wallpaper, a separate activity or the notification bar. Overall, 12531 gesture recognition attempts by 321 different users were recorded. Of these, 7263 (57.96 %) were performed on the lock screen, 3537 (28.23 %) via the notification bar, 1708 (13.63 %) via the activity, and 23 (0.18 %) on the wallpaper (see Figure 4). A similar distribution was observed for power users. Overall, the lock screen seems to be the preferred activator to launch applications, change settings or open a contact.

We further studied which activator is most frequently used per user. 135 users (42.06 %) most frequently used the lock screen to execute a gesture. 130 users (40.50 %) used the notification bar, and 56 users (17.45 %) used the activity most frequently. None of the users used the wallpaper as their most frequent activator. On average, 85.56 % (SD 15.16 %) of a users gesture recognition attempts were done using the same activator. This indicates that most users prefer to permanently use a single activator instead of various activators in parallel.

6164 of 12531 gesture recognition attempts, i.e., 49.19 % (48.22 % for power users), led to a successful recognition of a gesture. Thereby, the recognition rate varies between the individual activators, i.e., 34.78 % for the wallpaper and 67.01 % for the notification bar. We argue that the limited recognition performance can mostly be credited to users, who studied the limitations of the recognition, and accidental gesture recognition attempts, e.g., caused by unintentional lock screen or wallpaper use.

LAB STUDY

The study through the marketplace was conducted in an uncontrolled setting and its findings have high external validity. In the following, we complement these findings through a lab study, which will provide us with insights of internal validity. We study if lab users of Gestify create similar shaped gestures and what their preferred activators are.

Method

We recruited 18 participants (7 female) with an average age of 26.4 years (SD 5.20). The participants were recruited through the university's mailing lists. Twelve participants were students in different subjects such as Informatics, Electrical Engineering, etc. All participants owned an Android phone and 67 % used it for more than a year. None of the participants had used the Gestify application or another application with similar functionality prior to the study.

We installed Gestify on a Nexus 5 mobile phone and used it during the study to collect data. After welcoming and signing the consent form, participants were asked to fill in a demographic questionnaire. Then, we introduced Gestify through a short demo and explained its functionality. The participants were then asked to specify and train gestures for the 20 most popular actions, which we identified in the field study. The actions' order was randomized to reduce sequence effects. Finally, we asked the participants to answer a questionnaire, which assesses the participants' current and desired future use of activators for gesture shortcuts. Further, the questionnaire provided room to leave qualitative feedback about the participants' impressions of using Gestify. The study lasted approximately 30 minutes. Each participant was compensated with 5€.

Results

In total, 360 gestures were performed by the participants. However, because of a technical failure 7 gestures could not be recorded. Consequently, our analysis is done with 353 gestures. To have an overview on the gesture shapes we reviewed all gestures collected for each action and assigned them to one of the five shape categories, which we identified in the field study (see Figure 2). Based on these results, *letter*-shaped gestures were the most frequent gestures (35.41 %), followed by *icon*-shaped (26.91 %), *abstract*-shaped (26.62 %), *geometric*-shaped (9.06 %), and *word*-shaped (1.42 %) gestures. The two most frequent gestures for each action are shown in Table 2.

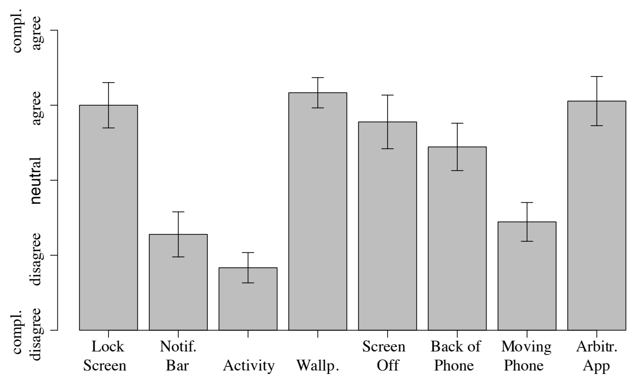


Figure 5. Lab study participants were asked to provide their level of agreement to a set of possible gesture activators. The results show that on average, participants mostly agree with performing gestures on the wallpaper. The error bars show the standard error.

In the questionnaire we asked the participants to rate a statement on each of the four different activators, e.g., *I would like to perform the gesture on the wallpaper*, using a 5-point Likert scale from (1) 'completely disagree' to (5) 'completely agree'. The analysis shows that 83.00 % of participants (mean 4.16, SD 0.85, median 4) would agree to use the wallpaper for gesture recognition. 77 % would agree to use the lock screen (mean 4, SD 1.28, median 4.5). 83 % of the participants disagreed with the statement on performing gestures in a separate application (mean 1.83, SD 0.85, median 2) as it—according to received qualitative feedback—would require more interaction steps and, thus, is cumbersome. Furthermore, 66 % disagreed with a gesture performance in the notification bar (mean 2.27, SD 1.27, median 2).

We further asked the participants to provide feedback on four other possible activators, i.e., the back of the phone, moving the phone itself, drawing gestures on the phone's screen when it is off, and draw a gesture while using arbitrary applications. The results reveal that 78 % of all users would agree to perform a gesture while using any arbitrary application (mean 4.05, SD 1.39, median 5). Further, 72 % agreed to perform gestures while the phone screen is off (mean 3.77, SD 1.51, median 4) and 56 % agreed with the back of the phone as an activator (mean 3.44, SD 1.33, median 4). 56 % disagree with moving the phone for performing a gesture (mean 2.44, SD 1.09, median 2). A visualization of these results is provided in Figure 5.

The qualitative feedback shows that the participants consider simpler gestures for frequently used actions, since these are easier to memorize. Furthermore, it was mentioned that existing symbols and icons, e.g., characters taken from the alphabet, can be a good starting point for the definition of gesture sets.

DISCUSSION AND RECOMMENDATIONS

In the following, we discuss our results and provide recommendations for designers and developers of future gesture recognition systems on mobile touchscreen devices, such as mobile phones.

Various Activators to Enable Gesture Recognition

Gestify offered four ways to start the recognition process, i.e., via lock screen, notification bar, wallpaper, or a dedicated activity. The field study indicated that the by far most frequently used gesture activator is the lock screen (credited for 58 % of all gesture executions, see Figure 4), followed by the notification bar (28 %), the activity (14 %), and the wallpaper (very close to 0 %). The lab study results show that 83 % would agree to use the wallpaper and 77 % would agree to use the lock screen as gesture activator (see Figure 5). In contrast, 83 % would disagree to use a separate activity as activator, and 66 % would disagree to perform gestures in the notification bar.

The lab study results indicate that users prefer the wallpaper activator, although the field study findings indicate that it was hardly used in practice. A potential reason for this gap could be that the field study users were not fully aware of the wallpaper activator, although it was equally well advertised as other activators. Another potential reason could be that the technique to enable the activator, i.e., a double tap on empty space on the wallpaper, was too complex to create a real advantage over using the regular launcher. Further, it could have been confusing or misleading that also the regular home screen keeps reacting to touch input during gesture execution, which can lead to minor icon movement.

Future gesture recognition systems should definitively respect the desire for diversity and offer various ways to recognize a gesture. For mobile phones, gesture recognition should be available through the lock screen, the notification bar, and through the wallpaper. However, because of our observations we recommend that the gesture interaction with the wallpaper needs further research. For example, it should be clarified how users exactly want an ideal gesture wallpaper to behave. Because we did not observe an intense use of the application activator and because people don't have a desire to use gesture recognition in a separate activity, this option could be omitted in future systems.

We further identified that the preferred activator is user specific. In about 85 % of all gesture recognitions, users use their most frequently used activator. For example, if a user's most frequently used activator is the lock screen, about 85 % of all gesture executions are done here. Thus, if for any reasons a user has to decide for a certain approach, most users will probably be able to make such a decision comfortably. Future research could investigate if gesture actions for individual activators differ, e.g., if messaging gestures are more often triggered from the lock screen or the notification bar.

Applications Are Most Popular Gesture Action

Gestify allowed users to use three types of gesture actions: *application gestures* to start applications, *contact gestures* to quickly reach for contacts, and *settings gestures* to modify system settings. In fact, *any* application and *any* contact could be assigned to a gesture. Further, Gestify provides a functionality, which—similar to existing quick settings applications—allows to instantly change system settings. Consequently, we argue that Gestify can be used to control these functionalities without any limitations.

Our results show that 85.10 % of all trained gestures in the field study were *application gestures*. Consequently, only a minority of *setting gestures* and *contact gestures* were created. This contradicts with earlier findings reported by Li [11], who observed that *contact gestures* (66 %) are preferred over *application gestures* (28 %). Even applications from the communication category and *contact gestures* together only sum up to 32.72 % of communication-related gestures, which is still half of the percentage that Li reported [11].

We argue that the reason for this difference lies in the slightly different interaction concepts. The app *Gesture Search* by Li consists of a single activity, which needs to be started and which then recognizes predefined character gestures. These are concatenated to words that serve as filter for a list of all available phone actions, such as 'start application' and 'show contact'. In contrast, Gestify can be accessed directly via various activators, allowing to trigger an action with a single user-defined gesture.

Thus, the omnipresence of gesture activators and the self-contained, 'single stroke' character of a gesture seem to be two key criteria for appreciated shortcut gesture recognizers. This is supported by our lab study, where users preferred to execute gestures, e.g., on the wallpaper, in every application or on the lock screen. All these activators are very present and often just a button press away. Further, we hardly observed *word*-shaped gestures in our studies. Instead, most users created letter, icon or abstract gestures.

Designers and developers should consider these findings in their future touch gesture recognition systems for shortcut access. Future research could investigate which of the proposed gesture activators perform best in everyday use. Further, also the execution speed of various gesture types could be studied and how this affects the overall user experience.

Pre-defined Gesture Set

We observed that many gestures were created for a small set of actions. In fact, we analyzed 1134 gestures from 388 users to identify a set of 20 popular actions. This set is shown in Table 2 and consists of 18 *application gestures*, one *settings gesture* and one *contact gesture*. We argue that this set is a good representation of the average user's preferred actions and, therefore, makes a good starting point for the definition of future shortcut tools for mobile devices.

Similar to Ouyang and Li [16], we observed that half of all 1134 gestures, i.e., 47.12 %, were *letter*-shaped. 35.89 % were of *abstract* shape, 8.73 % of *geometric* shape, 4.23 % of *icon* shape, and 3.00 % of *word* shape. This dominance is also present within our top 20 set of actions (see Table 2). Qualitative statements from the lab study indicate that some users used the alphabet and the action name as a reference for their gestures.

Overall, our findings show a clear preference for *letter*-shaped gestures. We argue that this is the case because they are powerful enough to represent and differentiate between advanced information, such as various apps. At the same time letters are simple enough to be memorized and remembered. In contrast, *abstract* and *geometric* gestures have a limited

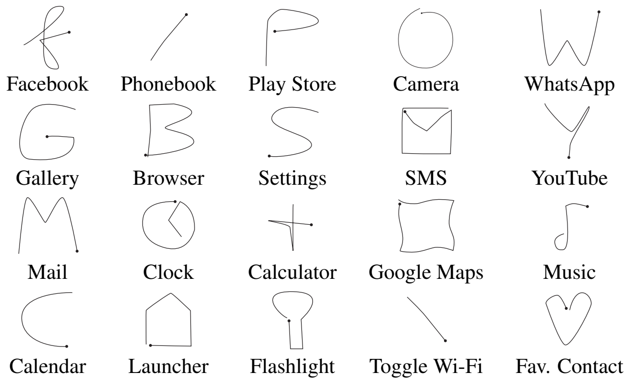


Figure 6. By analysing and combining gestures from both studies, we derived a comprehensive gesture set for the top 20 actions. This gesture set can serve as the default configuration for future mobile touch interfaces.

expressiveness, *icon* gestures can be too hard to remember, and *word* gestures can be too complex to draw.

Both, the identification of common actions and the subsequent analysis of gesture shapes, allow us to design a coherent and unambiguous gesture set. To derive this set we analyzed all gestures that we observed in both studies. We used the absolute number of gestures, which were observed in the field study, as a global indicator for importance. Starting from the most important gesture (left hand side in Table 2), we calculated an overall percentage for each identified shape by combining data from both studies. If the gesture shape with the highest percentage is not already used for a more important action, it was assigned to this gesture. Otherwise the next-most prominent shape is considered until an yet unused shape was found. This procedure allows us to consider both, i.e., how often a gesture was used, and which gesture shape was used most frequently.

The resulting set is shown in Figure 6 and could be used as a default working configuration, likely suiting the needs of most users. A deployment of this set could allow users to trigger actions without the need to train gestures in advance. This can positively affect the user experience and potentially reduce the time to learn and understand the gesture recognition application. Nevertheless, it must be clear that this set might also contradict the user's desires. Therefore, it should still be customizable to ensure greatest flexibility and to address users' individual needs.

Future research should study how gesture actions, shapes, and activators are designed for other user groups, e.g., users with a non-Latin alphabet, and if and how the availability of a default gesture set affects the user experience. Further, it would be interesting to know if the importance of individual actions change over time, and if and in which cases users re-define a gesture.

CONCLUSIONS

Touch gestures are becoming a more and more popular interaction technique on nowadays smart phones, tablets, and wearables. Existing touch gestures, e.g., a sideways slide to unlock a device, are typically defined by designers and developers, although it is unclear if these gestures meet the end-user requirements.

In this paper, we studied the users' needs and desires regarding gestures from the ground up. We designed and developed a flexible gesture recognition application, Gestify, which aims to fulfill most of the potential user needs. Users could define their own gesture actions, gesture shapes, and use different activators to initiate the gesture recognition. Using the Gestify application we conducted two exploratory user studies in two different settings: a field study in the Google Play marketplace and a supplemental lab study with 18 participants.

Our results show that the majority, i.e., 85 %, of all analyzed gestures are set up to control the user's applications. About 7 % each modify system settings or show contact information. Half of all gestures, i.e., 48 %, have a *letter* shape, typically mimicking the first letter of the controlled application name, e.g., *W* for *W*hatsApp. 36 % have an *abstract* shape, representing, e.g., a single swipe. Our field study results show that the lock screen (58 %) and the notification bar (28 %) are the most frequently used activators to trigger the gesture recognizer. However, the lab study reveals that the live wallpaper and back of the phone are also in the user's preference for triggering gestures.

We conclude that touch gestures are a valuable means for instant access to a variety of actions, e.g., to start applications, to show contact details, or to toggle system settings. We found that users prefer gesture activators, which are omnipresent and ideally less than a button press away. In fact, our results indicate that gesture execution should be possible on the lock screen, on the wallpaper, and in the notification bar. We observed that users rarely switch between various activators. Instead, they decide for a single activator and continue to use it. We argue that future gesture recognition systems should consider the need for diversity, but may ask users to decide for a single activator if necessary.

We further highlight the 20 most important actions which should be accessible through shortcut gestures. Of these 20 actions the majority are *application gestures*, such as Facebook, WhatsApp, or the Camera application. One action each is dedicated to access contact information or toggle system settings. We identify that about half of all gestures are *letter*-shaped, and the second most prominent gestures are of *abstract* shape. Overall, only a few gestures are of *icon*, *word*, or *geometric* shape. *Letter* shapes typically represent the first character of a related action, whereby *icon* gestures typically mimic the application icon. Through our two complementing studies we provide a comprehensive, unambiguous gesture set for the 20 most important actions (see Figure 6), which can serve as a customizable default gesture set for future touch devices and thereby avoids time-consuming gesture training efforts.

In our future work, we want to apply this proposed gesture set in the wild and study if users are satisfied or if they apply any modifications. Further, we are interested to investigate how gestures are used in everyday life. We plan to research which gestures are used most frequently over the day, and if certain activators are preferred to execute certain gestures. Further, it remains questionable to what extend our proposed gesture set can be transferred to future mobile devices, like wearable glasses or smart watches.

ACKNOWLEDGEMENTS

This work is partially supported by the German Research Foundation within the SimTech Cluster of Excellence (EXC 310/1).

REFERENCES

1. Anthony, L., Brown, Q., Nias, J., and Tate, B. Examining the need for visual feedback during gesture interaction on mobile touchscreen devices for kids. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, ACM (2013), 157–164.
2. Anthony, L., and Wobbrock, J. O. $n-protractor: A fast and accurate multistroke recognizer. In *Proceedings of Graphics Interface* (2012), 117–120.
3. Appert, C., and Zhai, S. Using strokes as command shortcuts: Cognitive benefits and toolkit support. In *Proceedings of CHI*, ACM (2009), 2289–2298.
4. Billinghurst, M. Gesture based interaction. In *Haptic input*, B. Buxton, Ed. Cambridge University Press, 2009.
5. Bolt, R. A. “put-that-there”: Voice and gesture at the graphics interface. In *Proceedings of SIGGRAPH*, ACM (1980), 262–270.
6. Döring, T., Kern, D., Marshall, P., Pfeiffer, M., Schöning, J., Gruhn, V., and Schmidt, A. Gestural interaction on the steering wheel: Reducing the visual demand. In *Proceedings of CHI*, ACM (2011), 483–492.
7. Henze, N., Pielot, M., Poppinga, B., Schinke, T., and Boll, S. My app is an experiment: Experience from user studies in mobile app stores. *Int. J. Mob. Hum. Comput. Interact. 3*, 4 (Oct. 2011), 71–91.
8. Hinrichs, U., and Carpendale, S. Gestures in the wild: Studying multi-touch gesture sequences on interactive tabletop exhibits. In *Proceedings of CHI*, ACM (2011), 3023–3032.
9. Lahey, B., Girouard, A., Burleson, W., and Vertegaal, R. Paperphone: Understanding the use of bend gestures in mobile devices with flexible electronic paper displays. In *Proceedings of CHI*, ACM (2011), 1303–1312.
10. Lee, J. C., Hudson, S. E., and Tse, E. Foldable interactive displays. In *Proceedings of UIST*, ACM (2008), 287–290.
11. Li, Y. Gesture search: A tool for fast mobile data access. In *Proceedings of UIST*, ACM (2010), 87–96.
12. Lü, H., and Li, Y. Gesture avatar: A technique for operating mobile user interfaces using gestures. In *Proceedings of CHI*, ACM (2011), 207–216.
13. Lü, H., and Li, Y. Gesture coder: A tool for programming multi-touch gestures by demonstration. In *Proceedings of CHI*, ACM (2012), 2875–2884.
14. Moyle, M., and Cockburn, A. Analysing mouse and pen flick gestures. In *Proceedings of the SIGCHI-NZ Symposium On Computer-Human Interaction*, ACM (2002), 19–24.
15. Nacenta, M. A., Kamber, Y., Qiang, Y., and Kristensson, P. O. Memorability of pre-designed and user-defined gesture sets. In *Proceedings of CHI*, ACM (2013), 1099–1108.
16. Ouyang, T., and Li, Y. Bootstrapping personal gesture shortcuts with the wisdom of the crowd and handwriting recognition. In *Proceedings of CHI*, ACM (2012), 2895–2904.
17. Roudaut, A., Bailly, G., Lecolinet, E., and Nigay, L. Leaf menus: Linear menus with stroke shortcuts for small handheld devices. In *Human-Computer Interaction – INTERACT 2009*, vol. 5726 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2009, 616–619.
18. Sahami Shirazi, A., Henze, N., Dingler, T., Pielot, M., Weber, D., and Schmidt, A. Large-scale assessment of mobile notifications. In *Proceedings of CHI*, ACM (2014), 3055–3064.
19. Schlömer, T., Poppinga, B., Henze, N., and Boll, S. Gesture recognition with a wii controller. In *Proceedings of TEI*, ACM (2008), 11–14.
20. Seyed, T., Burns, C., Costa Sousa, M., Maurer, F., and Tang, A. Eliciting usable gestures for multi-display environments. In *Proceedings of ITS*, ACM (2012), 41–50.
21. Tran, J. J., Trewin, S., Swart, C., John, B. E., and Thomas, J. C. Exploring pinch and spread gestures on mobile devices. In *Proceedings of MobileHCI*, ACM (2013), 151–160.
22. Wigdor, D., and Wixon, D. *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier, 2011.
23. Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined gestures for surface computing. In *Proceedings of CHI*, ACM (2009), 1083–1092.
24. Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *Proceedings of UIST*, ACM (2007), 159–168.
25. Wolf, K., McGee-Lennon, M., and Brewster, S. A study of on-device gestures. In *Proceedings of MobileHCI*, ACM (2012), 11–16.
26. Zhai, S., Kristensson, P. O., and Appert, C. *Foundational Issues in Touch-Surface Stroke Gesture Design*. Now Publishers Inc., Hanover, MA, USA, 2012.