Don’t touch that Tablet: An Evaluation of Gaze–based Interfaces for Tablet Computers

Milena Roetting
Hamburg University of Applied Sciences
Department of Computer Science
D-20099 Hamburg, Germany
Email: milena.roetting@gmail.com

Olaf Zukunft
Hamburg University of Applied Sciences
Department of Computer Science
D-20099 Hamburg, Germany
Email: zukunft@acm.org

***Abstract*—Gaze–based interaction has been proposed as an alternative to touch–based interaction for tablet computers. In this paper, we present Kucker, a system that implements different methods for processing gaze–based interaction on a tablet. Using Kucker, we perform an empirical usability test to evaluate different gaze–based interactions and compare them to the classical touch–based interaction. Our evaluation uses both user feedback through standard questionnaires as well as objective metrics like efficiency and effectiveness acquired during the usability tests. The results show that the efficiency of gaze–based interaction is inferior to touch–based interaction. We found out that users prefer some kind of gaze–based interaction to others. A classical touch–based interface is more efficient than any of the implemented gaze–based interfaces. However, gaze–based input is still considered as acceptable by users and can be used in special application areas or for special user groups.**

Keywords—Usability testing, User interaction, Interaction styles, Mobile computing, Vision I/O

I. Introduction

The widespread use of portable devices such as smart phones, eBook readers and tablets has created a market for daily life companions for millions of people around the world. The typical interaction with this kind of devices is through touch. Multi-touch gestures such as pinch to zoom are common to the users of these devices and are advertised as natural. However, according to studies like [1], there are still deficiencies regarding the usability of touch interactions on tablet computers. This especially holds true for tablets, a rising kind of computing devices. Therefore, alternative methods of interaction for tablet computers have been widely proposed in the literature, e.g. in [2], [3]. In this paper, we concentrate on gaze input for small screen tablet computers. Our goal is to evaluate the usability of existing gaze–based input methods for typical interactions. For this, we design and implement Kucker, a gaze–based interaction application for tablet computers. Kucker provides several different gaze–input methods and can be used for both gaze-only and combined gaze-and-touch gestures. Using Kucker, we compare different gaze–based interaction style proposed in the literature with the traditional touch–based interaction using an experimental setting. Here, we use a task-based approach common to usability-testing to evaluate the feasibility of gaze–based interactions. Finally, we discuss the results of our experiment and give a conclusion.

II. Related Work

There exists a number of approaches for using gaze–based input as an interaction method for traditional desktop computers. They all have to solve one of the main problems of gaze-based interaction, namely to distinguish between gazes meant to “read” (or collect) data presented on the screen and gazes that are meant to give input to the computer. To solve this so called "Midas-touch problem", three approaches are presented in [4]:

1) Blinking while looking at the interaction object
2) Dwelling on the interaction object, i.e. to look longer than a predefined threshold
3) Performing an additional action like pressing a button while gazing at the interaction object.

Additionally, special gaze–gestures have been proposed as a method to solve the Midas–touch problem. In [5], Kudos et al. measure the divergence in eye movement, thereby enabling users to “press” a button by moving their viewpoint forward. In the smooth pursuits approach [6], [7], object selection is based on gaze–tracking objects moving on unique trajectories. By recognizing the unique trajectories, a best match to the trajectories of the moving objects is performed.

Current research like [3] and first products like Samsung SmartScroll™ present an approach where the user gaze is tracked without an eye-tracker. Instead, the build–in camera is used to track the gaze using modified image–recognition algorithms implemented on the tablet. However, no usability study about the acceptance of gaze–input is included. On the other hand, our evaluation is independent of the used gaze–technology and therefore also applicable to these new technological approaches.

In [2], Dybdal et al. compare different methods of gaze input, namely long–looking and gestures on a smart phone. Compared to our approach, their approach is limited in scope by evaluating "selection of objects" only and by using a small–screen smart phone which prohibits dedicated touch–zones on the screen. Additionally, they also limit their methods of gaze–input to two different methods while we also include trajectories in our study and examine the learning effects of volunteers. The work of Holland et al. ([8]) focuses on the evaluation of the eye-tracking calibration only and is therefore — despite of the title — not comparable to our work.

MobiCASE 2014, November 06-07, Austin, United States

DOI 10.4108/icst.mobicase.2014.257760

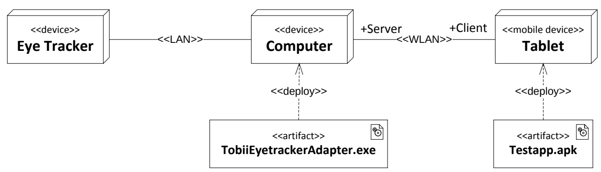

Fig. 1. Kucker: Deployment diagram

III. The Kucker–System

The Kucker-system is designed to allow experiments with touch-based, gaze-based and multi-modal interactions between a user and an unmodified tablet computer.

A. The technology–view of Kucker

The Kucker–system is installed in an industry-grade usability lab and combines the Kucker software with an eye tracker, a tablet and integrates the facilities available in the usability lab like cameras and microphones. We currently deploy a Tobii eye-tracker X120 to allow gaze-input to the Kucker-system. Figure 1 gives an overview of the Kucker architecture. The gaze input from the eye–tracker is processed on a server computer that connects via LAN to the eye-tracker and via WIFI to the tablet computer. The server basically processes the gaze data and forwards them to the Kucker test application running on the tablet. In the current setting, we are able to transfer 60 gaze–datasets per second from the eye-tracker via the server to the tablet. The tablet gets the gaze input and the Kucker test application maps the gazes onto input signals of the underlying operating system. The current implementation of Kucker uses the Android OS for the tablet test application and the Tobii mobile stand as a hardware component to integrate the tablet and the eye-tracker (see figure 2). In this installation, the user is not able to move the tablet. Hence, we do not evaluate problems specific to mobility. Nevertheless, all problems that arise in this lab setting will also occur in the mobile scenario, while additional problems may arise.

The Kucker test application allows to process different user-defined gestures. Currently, our design manages ten different interactions and for each interaction it provides at least two different gestures (touch and gaze). Figure 3 shows the

Fig. 2. Kucker: Mobile stand (Source: tobii.com)

component diagram of Kucker. The component "Testprocess" guides the user through the tests and uses the tasks which are defined and configured in a separate component. It writes its data into another component which can be used for the analysis of test data. The testprocess further uses two different components for the touch– and gaze–interaction and their respective tasks. The implementation for the server and the test application is done in Java and the source–code is available for research purposes on request.

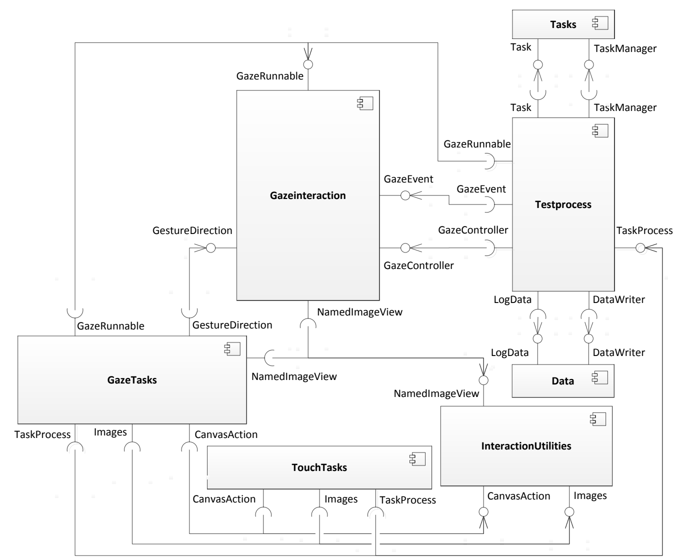

Fig. 3. UML–Component diagram of Kucker

B. Gaze– and Touch–based Interaction in Kucker

Kucker is designed to perform experiments with different gaze– and touch–based interactions. It easily integrates new interaction forms into the system. For the experiment described in this paper, we have implemented the following different gaze–based interactions:

- Dwelling at an object
- Gaze–based gestures
- Gazing and pressing a button concurrently
- Smooth pursuits.

For each of these different interactions, we ask volunteers to perform tasks using these interactions. The selection and manipulation of objects are the most common tasks for the user of a tablet and are represented in the test application of the Kucker–System by the following tasks (grouped by interaction):
Simple selections: The following are very simple tasks typically executed by tablet users

1) Start an app
2) Cut, copy and paste an object
3) Open an object
4) Undo an operation

Complex selections: A special case is the selection of multiple objects, which is not as common as the interactions above and which is typically considered as an advanced interaction

5) Selection of multiple objects

Object manipulation:

6) Move an object
7) Scrolling within an object screen

Gestures: Finally, we integrated the following interactions that are implemented by gestures which typically have to be learned by the user of a tablet

8) Close an app
9) Navigate back in an interaction
10) Delete an object

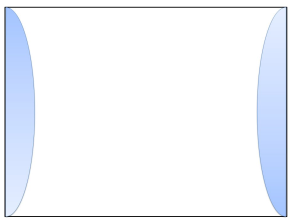

Fig. 4. Buttons for gestures (left) and selections (right) at the side of the tablet screen for multi–modal interaction

In order to solve the Midas-touch-problem, we decided to integrate various interaction techniques known from the literature into Kucker. First, a multi-modal interaction is integrated. As illustrated in figure 4, the test application of Kucker provides two additional touch-buttons at the left and the right side of the tablet. The button on the right is used to confirm a selection done by a gaze. I.e. after looking at an object, the user presses this button with his thumb and therefore confirms that the object he is looking at should be selected. This allows to distinguish between the gaze–based selection and a simple "look at the object". The button under the left thumb of the user is pressed to indicate that the user is performing a gaze gesture. When pressed, Kucker interprets the gaze as one of the three implemented gestures, depicted in figures 5, 6 and 7

Fig. 5. Gaze gesture for navigating back

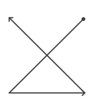

Fig. 6. Gaze gesture for deleting an object

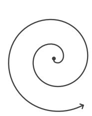

Fig. 7. Gaze gesture for closing an app

The task 1 "start an app" is implemented with a combination of gaze and touch input. It starts the app the user is look at as soon as he is pressing the button under his right thumb. The task 3 "open an object" is – although quite similar – implemented using another interaction technique. Here, we use the Smooth Pursuits-based interaction technique. Basically, we move each object on a specific trajectory and as soon as a user follows this trajectory longer than a predefined threshold, the object in opened. Consequently, the test application recognizes the different trajectories and maps them onto the associated objects.

The tasks 5 "selection of multiple objects" and 2 "cut, copy and paste" are implemented with a third kind of interaction technique. Here, we use the dwelling, i.e. the long gaze at an object. For the multiple selection, a long gaze at one of multiple objects selects it and allows selecting other objects. For cut, copy and paste, a long gaze at an object activates the common context menu for these operations. Using the unique opportunities of eye-tracking, we choose to implement the tasks 8, 9, and 10 "close an application", "navigate back", and "delete" by varying gaze gestures. We designed different customized gazes that should have different complexities in executing them. Figure 5 shows the "back"-gaze–gesture. This is the least complex gaze gesture. It is motivated by the corresponding touch-gesture on iOS-devices. Figure 6 shows the "delete"-gaze–gesture. This one is meant to remind the user of a cross that is found on windows-UI and is constructed using angled lines. Finally, the "close app"-gaze–gesture shown in figure 9 is the most complex one and is expected to be rather disturbing to users. It explicitly uses no angled lines but circular forms.

The task 7 scrolling is implemented by creating a dedicated area on the top and bottom of the screen which, when gazed at, scrolls the screen. This is the most intuitive interaction form for scrolling ([9]). Finally, we implemented the interaction 4 and 6, "undo" and "move an object" similar to the concept know from the touch interface. Following a long gaze at an object, a context menu pops up and when "move" is selected, the object follows the gaze to the new position. When "undo" is selected from the context menu, the previous manipulation is undone.

IV. Evaluation

In this chapter, we want to present an empirical study to answer the following research questions:

1) Is gaze-based interaction suited for the average user when compared with touch-based interaction?
2) Are there gaze-based interactions where the use of them is as efficient as touch-based interactions?

A. *Evaluation metrics*

In order to answer these questions we perform an empirical study using Kucker. To quantify the findings, we use the following metrics for different aspects of the interactions in Kucker:

Execution time:
The time a volunteer needed to complete a given task.

Error rate:
The number of errors done by a volunteer for

a given task. The possible errors are defined together with the tasks to be performed.

Number of trials:
How often does a volunteer restart a given task, for example by rereading the task description.

Based on these three metrical values, we compute the effectiveness and efficiency of the different interactions using the following formulae:

$$\text{effectiveness} = \frac{1}{\left(\frac{G_V}{G_V+G_F} * \text{\#Trials}\right) + \left(\frac{G_F}{G_V+G_F} * \text{ErRate}\right)} \quad (1)$$

$$\text{efficiency} = \frac{\text{effectiveness}}{\text{execution time}} \quad (2)$$

The formula 1 was chosen so that a weight can be assigned to the number of errors or number of trials. G_V is the weight of the trials, G_F the weight of the errors, ErRate the error rate and #Trials the number of trials.

In addition to these metrics, we used a questionnaire to measure the user experience and get subjective feedback from the volunteers. We adopt a combination of the After-Scenario Questionnaire (ASQ) ([10]) and the System Usability Scale (SUS) ([11]). The ASQ asks the following three questions that will be answered for each scenario on a Likert-scale with seven elements:

1) Overall, I am satisfied with the ease of completing the tasks in this scenario.
2) Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario.
3) Overall, I am satisfied with the information when completing the tasks.

These three questions cover the fundamentals aspects of usability research, namely effectiveness, efficiency and satisfaction. ([12]). However, since all questions are expressed positively, the volunteer may be biased. This disadvantage is not present in the SUS questionnaires. There, half of the questions are expressed positive and half are expressed negative. Here, we use the following ten questions that are answered on a Likert-scale with five elements:

1) I think that I would like to use this system frequently.
2) I found the system unnecessarily complex.
3) I thought the system was easy to use.
4) I think that I would need the support of a technical person to be able to use this system.
5) I found the various functions in this system were well integrated.
6) I thought there was too much inconsistency in this system.
7) I would imagine that most people would learn to use this system very quickly.
8) I found the system very cumbersome to use.
9) I felt very confident using the system.
10) I needed to learn a lot of things before I could get going with this system.

According to [13], SUS provides valid data starting for groups of eight volunteers.

B. Test Procedure

The test is structured in three parts: The introduction, a main part and an optional supplement. In the introduction, we explain the test objectives to the volunteers and collect a consent form from each volunteer. Also, we collect demographic information about age, gender, previous knowledge etc.

In the main part of the test, different tasks are performed by each volunteer using the tablet and afterwards the questionnaire is done. The test is recorded using five cameras (see figure 8). Each task is performed twice. First, the volunteers perform the task using the traditional touch-interface. This is used not only to gain data for comparing with gaze-based interaction, but also to introduce the volunteers to the setting and the tasks. Afterwards, all tasks are performed again, but this time using the gaze-interface. Using this order, the volunteers were able to get familiar with the tasks before the gaze–based interaction. Since all volunteers had former experience with touch–based interaction and non had experience with gaze–based interaction, we used this order to generate a "one new thing per time" situation. First, the task was new for the volunteers. Than, the gaze–based interaction was new for them.

We used the Tobii eye tracker model X120 and performed the tests on an Asus tablet with a 10" screen running Android 4.

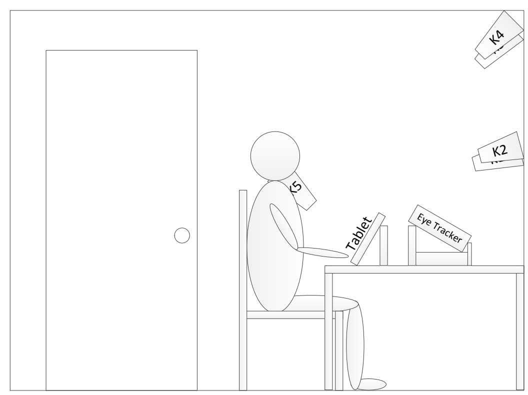

Fig. 8. Test setting

The volunteers are asked to perform 10 different tasks covering all the interactions described above. Since each task is performed once by touch and once by gaze, the volunteers perform 20 tasks. Additionally, we retest 2 tasks to measure the learn-ability of gaze–based interaction. After finishing each task, the ASQ-Questionnaire is filled out. After finishing all touch resp. gaze-tasks, the SUS-questionnaire is filled out for each interaction form. Additionally, we encourage the volunteers to “think-aloud”. These comments are synchronized together with the video-data and are rendered into an audio-video-presentation.

The main part of the test is structured and guided using the test application of Kucker. In addition to the guidance provided by the app, the volunteers are provided with a brief text describing the following task. This text is read before the task is started and can be read again anytime. Every re–read is registered by the test director and counted for the "number

TABLE I. DEMOGRAPHIC ATTRIBUTES OF THE VOLUNTEERS

Feature	**Volunteer**							
	1	2	3	4	5	6	7	8
Gender	m	m	f	m	f	f	f	m
Age	26-40 Years	<26 Years	<26 Years	26-40 Years	>40 Years	<26 Years	26-40 Years	26-40 Years
Experience with tablets	Very good	None	Good	Very good	None	Aver–age	Good	Very good
Background in CS	Yes	Yes	No	Yes	No	No	Yes	Yes

of trials". The first task is the opening of an app, which is the easiest task in both the touch and the gaze–based realization and therefore well suited to introduce the volunteer to the test. At the end of the test, the volunteers are interviewed and can optionally provide additional feedback to the test director.

C. Volunteers

The characteristics of the group of volunteers that agreed to use the Kucker–System and to participate in the usability–study is shown in table I. We asked eight volunteer to participate in the study. According to [13] and [14], eight is a number of volunteers where results can be considered valid.

D. Test tasks

The following ten tasks were tested using both touch-based and gaze-based interaction.

1) Open an app : An element or button is selected

Task: Please open the picture-app. Please open the canvas-app.
Touch-based interaction:
Touch the associated item.
Gaze-based interaction:
Look at the associated icon and touch the gaze–button on the side of the tablet to confirm the selection.

2) Scrolling: The content of a page, which is currently not visible, is shown.

Task: Please get an overview of all photographs.
Touch-based interaction:
Wipe with a finger into the opposite direction of the hidden content.
Gaze-based interaction:
Look at the scrollbar at the end of the screen.

3) Open an object : Open the detail view of an object, here an email or a photo.

Task: Please choose the photo with the cat
Touch-based interaction:
Touch the object in the grid–based view.
Gaze-based interaction:
The objects move in different and unique trajectories. One object is selected by gazing at it so that the trajectory can be mapped onto the object, which is subsequently opened.

4) Move back : Navigation within an app to the last visited page.

Task: Please move back to the grid–based view of all photographs.
Touch-based interaction:
Wipe the finger from right to left on the screen
Gaze-based interaction:
Touch the gesture–button on the left of the screen an perform the "back gaze–gesture" (see figure 5)

5) Multiple selection: Multiple objects are selected and can subsequently manipulated together

Task: Please select a given number of photos.
Touch-based interaction:
Touch one photo longer than usual, select the select–button from an up–popping context menu and then touch more photos shortly.
Gaze-based interaction:
Dwell at every photo that should be selected

6) Delete: Delete one or more selected objects

Task: Please delete the photo with the cat.
Touch-based interaction:
The photo is selected through dwelling and then a delete–button from a context–menu is selected
Gaze-based interaction:
The photo is select through a long look at it and then the predefined delete–gaze–gesture (see figure 6) is performed while holding the gesture–button on the left

7) Closing an app: An app is closed and the home–screen is displayed

Task: Please close the app
Touch-based interaction:
Press the home–button
Gaze-based interaction:
The predefined close–gaze–gesture (see 7) is performed while holding the gesture–button on the left

8) Cut, copy and paste: An object or a selected text on the screen is to be moved by copy and paste.

Task: Please copy an element and insert it at some other position
Touch-based interaction:
Open the context menu by a long touch on the element and select the copy–entry from the context menu by touching it.
Gaze-based interaction:
Dwell at the element to open the context menu and select the copy–entry from the context menu by looking at it.

9) Move an object: An object is moved on the screen

Task: Please move the picture with the cat
Touch-based interaction:
Touch the object in question and hold it. Then, move it to the new position

Gaze-based interaction:

Look at the object in question long. At the context menu, select the move entry. Then, move the object by looking at the new position. When the final position is reached, touch the button on the right side.

10) Undo: Undo the manipulation of an object and restore the former state.

Task: Please move an object on the screen and undo the movement

Touch-based interaction:

Touch the object longer and select the undo–entry in the up–popping context menu by touching it

Gaze-based interaction:

Look at the object longer and select the undo–entry in the up–popping context menu by looking at it

These ten tasks cover all the capabilities of the Kucker–system and use the gaze–concepts for mobile users as described in section II.

E. Test results

The results from our tests are presented in this section. As described above, we have used the SUS–Value, which generates basically one value for the whole interaction–style, and the After–Scenario–Question (ASQ–value), where we gain one value per task. Figure 9 shows the average SUS–value

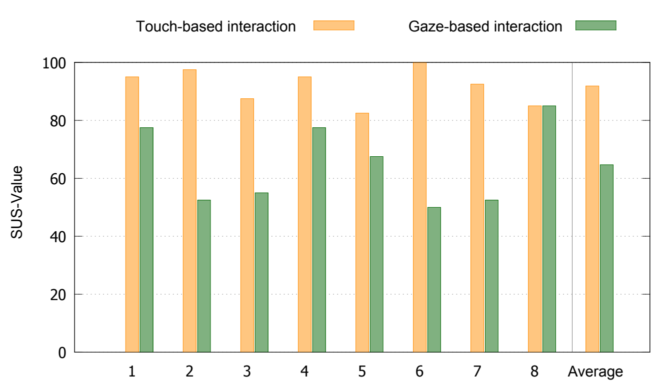

Fig. 9. SUS-Value per volunteer and overall for touch–based and gaze–based interaction

ordered by volunteers. The average touch–based SUS–value is 91.875 and it is 64.6875 for the gaze–based interaction. The value for the gaze–based interaction is therefore significantly lower. According to empirical studies of 50 systems [12], an SUS–value above 60 is considered acceptable, while an SUS–value of above 90 is extraordinary high.

With ASQ, the usability of a single task is evaluated. The first of the three statements describes the effectiveness of the scenario, the second the efficiency and the third uses a combined statement to describe the overall satisfaction of the users in the task in question. The volunteers are asked to give their opinion on a scale between 0 ("I do not agree at all")

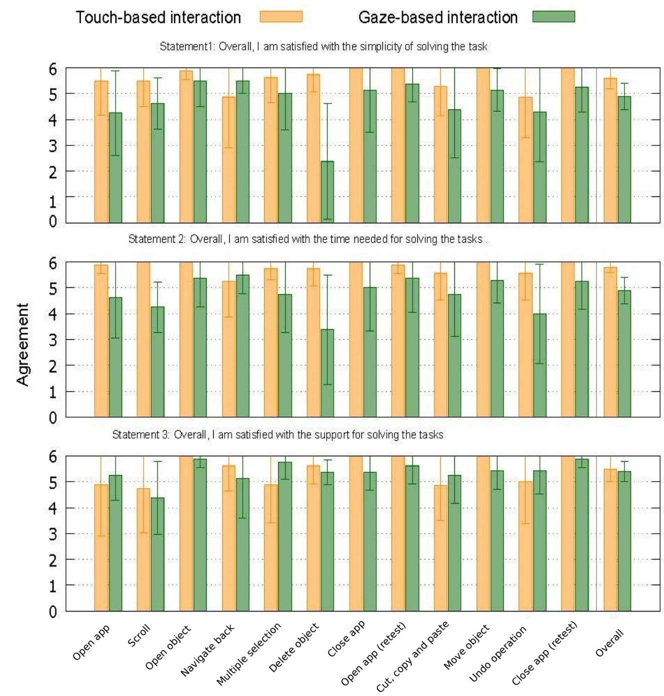

Fig. 10. Mean ASQ-Value and deviation per volunteer / overall and test for touch–based and gaze–based interaction

and 6 ("I fully agree"). The test results for ASQ are depicted in figure 10 using the average of all volunteers.

It can be seen that the results for all tasks using touch–based interaction are in the range of 4 to 6, i.e. the volunteers are very satisfied using this interaction style. Regarding the gaze–based interaction, the values are always lower than the corresponding touch–based interaction. Nevertheless, they are still in the average range and far from catastrophic results.

While SUS and ASQ are subjective metrics based on the user experience of the volunteers, we also collected objective metrics. As described in section IV-A, we computed the effectiveness and efficiency of the volunteers using the data acquired while performing the test. These are based on the execution time, errors and repeated executions of tasks. We collected the data using the Kucker test application (execution time) and the camera–based recordings of the tests (errors and re-execution).

The two figures 11 and 12 show the data for effectiveness and efficiency. All errors and re–executions we given the same value. (see formula 1).

With the exception of the task "scrolling", the touch–based interaction is always more efficient than the gaze–based interaction and it is without exception always more effective. We discuss the unique characteristic of scrolling as an non–precision interaction in section V.

The tasks "open app" and "close app" are executed twice during the test. The motivation is to get data about the learning effect especially for gaze–based interaction. In order

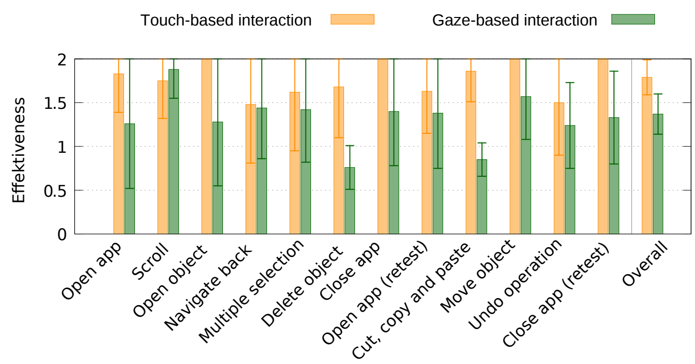

Fig. 11. Mean effectiveness and deviation per test task and overall

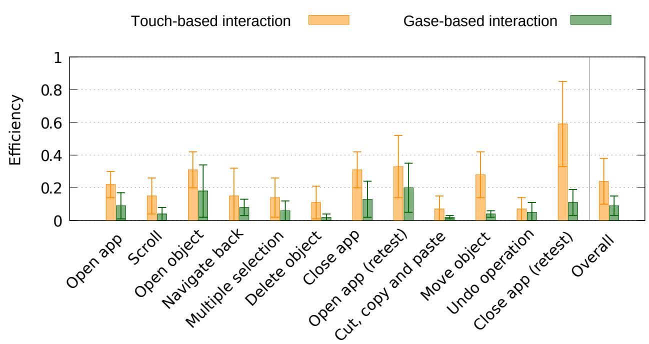

Fig. 12. Mean efficiency and deviation per test task and overall

to not annoy the volunteers, we use different applications to be opened/closed. Nevertheless, the tasks are similar. The execution time of the tasks "open app" and "close app" are both notable smaller in the second run. The data is shown in II and discussed in the next section.

V. Discussion

For the gaze–based interaction, three different methods of selection have been evaluated. Based on the experiment, smooth pursuits are well suited for the selection of an element from a set of similar elements. Regarding the selection of a simple object like a button, i.e. the most typical selection process in an UI, we have to distinguish between efficiency and satisfaction of volunteers. According to the feedback gained through questionnaires, the selection by dwelling is favored by the volunteers. According to the efficiency–metrics, the selection by "look at object and press the button" is in fact faster than the dwelling variant.

TABLE II. Execution time for repeated tasks

Task	**Touch–based interaction**		**Gaze–based interaction**	
	Average [s]	**Median [s]**	**Average [s]**	**Median [s]**
open app	8,50	8,00	19,50	16,00
open app (Retest)	5,00	4,50	11,25	7,50
Close app	6,50	6,00	23,25	17,00
Close app (Retest)	3,38	3,00	17,75	12,50

TABLE III. Evaluation of the task "navigate back" (gaze gesture)

	Touch–based interaction			**Gaze–based interaction**		
	Avg.	**Median**	**Standard deviation**	**Avg.**	**Median**	**Standard deviation**
ASQ - Statement 1	4,88	6,00	1,96	5,50	5,50	0,50
ASQ - Statement 2	5,25	6,00	1,39	5,50	6,00	0,71
ASQ - Statement 3	5,63	6,00	0,99	5,13	6,00	1,54
Duration [s]	9,88	7,50	6,70	20,63	17,00	10,53
Effectiveness	1,48	2,00	0,67	1,44	1,50	0,58
Efficiency	0,15	0,24	0,17	0,07	0,06	0,05

TABLE IV. Evaluation of the task "open object" (smart pursuit)

	Touch–based interaction			**Gaze–based interaction**		
	Avg.	**Median**	**Standard deviation**	**Avg.**	**Median**	**Standard deviation**
ASQ - Statement 1	5,88	6,00	0,33	5,50	6,00	1,00
ASQ - Statement 2	6,00	6,00	0,00	5,38	6,00	1,11
ASQ - Statement 3	6,00	6,00	0,00	5,88	6,00	0,33
Duration [s]	6,38	6,50	1,87	11,38	9,00	7,05
Effectiveness	2,00	2,00	0,00	1,28	1,33	0,73
Efficiency	0,31	0,31	0,11	0,11	0,14	0,16

We also implemented gaze–gestures as a mean to initiate the manipulation of objects. The experiments show that gaze–gestures can be performed effectively. However, the volunteers did not consider them intuitive or simple in their execution. When comparing the three different gaze–gestures, we found that the gesture for "navigate back" (see table III) is the most acceptable one. The two other gestures are considered more complicate since they require different gaze directions. We conclude that gaze gestures should preferably make use of single–direction patterns. Even then, the efficiency of gaze–gestures is approx. half of a classical touch–based gesture (see table III). In order to recognize gestures and to distinguish them from classical reading of screen objects, the gesture button implemented in Kucker has been a good choice. This button solves the Midas-touch problem independent of the gaze–gesture.

The application of gaze–based interactions has to take into account the precision required for the different tasks. When comparing the tasks "scrolling" and "move an object", we notice that the precision required for moving an object to a new position is higher than for scrolling of text or a map. Since the precise execution of gaze–based manipulation is rather inefficient and exhausting, a gaze–based interaction is especially useful for non–precise interaction tasks like scrolling while the execution of high–precision interaction tasks should be preferable done using touch-interaction.

The application of smart pursuits, i.e. the following of unique trajectories, is tested for the "open an object" task within Kucker. Here, we notice the biggest difference in the evaluation of the gaze–based interaction by the volunteers and the efficiency metrics. As shown in table IV, the volunteers are very satisfied with the gaze–based interaction based on the ASQ–value. This is contrasted by the values for effectiveness, efficiency and duration, that show that the gaze–based execution of the task take significantly longer and is more error-prone than the touch–based interaction. This is probably due the fact that the volunteers were rather enthusiastic about this kind of gaze–based interaction and rated it almost as good as touch–based interaction.

All volunteers had some kind of experience with touch–interaction. As shown in table I, some of them consider themselves as "very well experienced" with tablet computers. But also the volunteers with no previous experience using tablet computers had already seen other people using touch–interfaces. In fact, we were unable to recruit any volunteers without prior exposition to touch–interfaces. This can of course bias the evaluation. Hence, we include the retest of some gaze–based interaction to find out if the repeated execution of gaze–gestures has an effect on the efficiency and effectiveness. We noticed that the learn-ability of gaze–gestures is substantiated by the data shown in table II and further supported by the data for the second retest (not presented here due to space constraints). Further research will be needed to quantify the learn-ability effects of gaze–based interaction.

Generally, the measured values for efficiency and effectiveness show a bigger difference than the data collected using the questionnaire. Hence, the volunteers are more open to gaze–based interaction compared to the measured values which show bigger disadvantage for the gaze–based interaction.

VI. Conclusion

We presented Kucker, a system for experiments with various methods for gaze–based interaction. Using Kucker, we successfully evaluated existing proposals for interacting with tablets not by touch–based interaction but gaze–based interaction. Our research was guided by research questions presented in section IV. While existing research often focuses on new ways to implement gaze-interaction, we found out that existing approaches to gaze–based interaction are typically less efficient, less effective and less satisfying for users than touch–based interactions. Consequently, we propose to focus further research on new application areas for gaze–based interaction. This could be for handicapped people or for application areas like maintenance where one or both hands are used for other tasks than interacting with the tablet. Furthermore, as eye-tracking technology matures, we will conduct further studies including mobile users.

References

[1] J. Nielsen, "Tablet usability," Nielsen Norman Group, August 2013, downloaded: 08.08.2014. [Online]. Available: http://www.nngroup.com/articles/tablet-usability/

[2] M. L. Dybdal, J. S. Agustin, and J. P. Hansen, "Gaze input for mobile devices by dwell and gestures," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '12. New York, NY, USA: ACM, 2012, pp. 225–228.

[3] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '14. New York, NY, USA: ACM, 2014, pp. 207–210. [Online]. Available: http://doi.acm.org/10.1145/2578153.2578185

[4] R. J. K. Jacob, "What you look at is what you get: eye movement-based interaction techniques," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '90. New York, NY, USA: ACM, 1990, pp. 11–18.

[5] S. Kudo, H. Okabe, T. Hachisu, M. Sato, S. Fukushima, and H. Kajimoto, "Input method using divergence eye movement," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '13. New York, NY, USA: ACM, 2013, pp. 1335–1340.

[6] M. Vidal, A. Bulling, and H. Gellersen, "Detection of smooth pursuits using eye movement shape features," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '12. New York, NY, USA: ACM, 2012, pp. 177–180.

[7] M. Vidal, K. Pfeuffer, A. Bulling, and H. W. Gellersen, "Pursuits: eye-based interaction with moving targets," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '13. New York, NY, USA: ACM, 2013, pp. 3147–3150.

[8] C. Holland, A. Garza, E. Kurtova, J. Cruz, and O. Komogortsev, "Usability evaluation of eye tracking on an unmodified common tablet," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '13. New York, NY, USA: ACM, 2013, pp. 295–300. [Online]. Available: http://doi.acm.org/10.1145/2468356.2468409

[9] M. Kumar, T. Winograd, and A. Paepcke, "Gaze-enhanced scrolling techniques," in *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '07. New York, NY, USA: ACM, 2007, pp. 2531–2536. [Online]. Available: http://doi.acm.org/10.1145/1240866.1241036

[10] J. R. Lewis, "An after-scenario questionnaire for usability studies: Psychometric evaluation over three trials," *SIGCHI Bull.*, vol. 23, no. 4, pp. 79–, Oct. 1991.

[11] J. Brooke, "Sus - a quick and dirty usability scale," Redhatch Consulting Ltd., Tech. Rep., 1986.

[12] T. Tullis and B. Albert, *Measuring the User Experience - Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann Publishers, 2008.

[13] T. S. Tullis and J. N. Stetson, "A comparison of questionnaires for assessing website usability," Fidelity Investments, Tech. Rep., 2004.

[14] J. Nielsen, *Usability Engineering*. Mountain View, California: Morgan Kaufmann, 1993.