The Link-Offset-Scale Mechanism for Improving the Usability of Touch Screen Displays on the Web

Willian Massami Watanabe, Renata Pontin de Mattos Fortes, and Maria da Graça Campos Pimentel

Computer Science Department, Institute of Mathematics and Computer Science at University of Sao Paulo, Sao Carlos, Brazil
{watinha,renata,mgp}@icmc.usp.br

Abstract. Touch-screen interfaces have become a widespread-input-device tendency for computer systems. In this context, many studies investigate how to improve general usability for touch-screen devices. These studies consider different interaction design features that improve the usability for touch sensitive surfaces, considering the low accuracy it presents, given obstacles such as the "fat finger problem", low-perception of pointing mechanisms, difficulties in the selection of small objects, among others. This work aims at presenting the link-offset-scale touch interaction mechanism for improving the usability for touch-screen devices. The link-offset-scale mechanism makes use of web-application-structure meta-data (identifying links) to provide feedback information about the selection of links in touch interfaces, while the surface is touched by the user. The link-offset-scale mechanism's primary goal is to reduce the number of errors that users commit while interacting with touch-screen devices in the Web.

Keywords: Touch-screen displays, pervasive computing, ubiquitous computing, usability in touch-screen devices, web usability.

1 Introduction

Touch-screen devices are becoming more and more frequent in personal computers, airports, banks and other environments [11,20]. Abowd et al. [1] describe touch screen devices as direct input mechanisms, that fit Pervasive Computing environments by providing natural interaction between the user and the computer. Touch screen interfaces are classified as "direct" input devices because of the integration between interface control and other information in a single surface [17]. The unification of input/output, control/feedback, hand gestures and eye movements suggests to researchers the idea that touch screen displays are intuition-oriented, particularly for non-frequent computer users [14].

There have been many studies which approach how to improve general usability for touch screen devices [8,9,11,12,18,20]. These studies consider different interaction design features that improve the usability for touch screen displays, considering the low accuracy it presents given the "fat finger problem", low-perception of pointing

P. Campos et al. (Eds.): INTERACT 2011, Part III, LNCS 6948, pp. 356–372, 2011.

mechanisms, difficulties in the selection of small objects, among others [17]. It is worth noticing that these studies present design principles that should be addressed by interface designers.

On the other hand, the large capacity of Web for providing information accounts for multiple possibilities and opportunities for users. The development of high performance networks and ubiquitous devices allow users to retrieve information content from any location and in different scenarios or situations they might come across in their lives. Further to retrieving information, current technologies also enable users to contribute to the authoring of content on the Web by means of forums, blogs and wikis, in the so called "Web 2.0" [19]. However, a great amount of effort would be required to redesign all web applications now available in order to make them conformant with these design principles, and thus make the web more usable for touch screen displays.

In this context, this work aims at elaborating an automatic adaptation strategy for web application presentations in order to improve their usability when users are interacting with touch screen devices. Currently, the Web presents content in HTML, XML, JSON, among other formats. All these data are presented through web browsers that abstract the different platforms [4]. With the establishment of Ajax [6] and the evolution in the HTML5 profile specification [7], there has been a significant improvement in the interactivity of web applications. And, in this context, this work has the objective of making use of these newly evolved technologies (Ajax and HTML5) to maximize usability for touch screen displays, by implementing functionalities that automatically adapt the presentation of web applications to improve the general interactivity of websites for touch screen inputs.

The key functionality implemented in this paper is named link-offset-scale. The functionality provides feedback as soon as users have their finger over link elements in a webpage. The feedback presents a larger version of the link content (scaled) and is positioned in a different location to the link itself so as to be visible to the user, but not at the same position where his/her fingers are on the touch device. As the user stops touching the link, the link-offset-scale mechanism redirects him/her to the webpage the link referred to. This mechanism's goal is to improve the usability of touch screen devices by increasing the accuracy and usability of touch interactions for web applications already available. In order to validate the described method, we conducted usability experiments on web applications, comparing accuracy and task realization timing metrics between users interacting with and without the resource.

This work is structured as follows: next section (Section 2) presents the state-of-art studies on touch interaction, which aim at improving accuracy of touch devices; Section 3 describes our design solution for improving usability of touch displays in the Web; Section 4 details a usability experiment carried out to measure the performance of our proposal; Section 5 presents a discussion about the results of the experiment; and finally, Section 6 presents some final remarks and future works.

2 Related Work

Even though touch screen displays are classified as intuition-oriented and easy to learn input devices, they present several limitations to be overcome by the user [17], which include:

1. Difficulty in the selection of small objects;
2. Problems related to the visual occlusion of elements and to imprecision and lack of calibration of the device;
3. And the non-existence of an analogue touch interaction for every possibility of mouse interaction.

These limitations are related to the accuracy of touch interactions with the display. Holz and Baudisch [8] explain that touching a small target represents an error prone activity that can be observed by having users acquire a small target repeatedly. The more inaccurate the device, the wider the spread of distribution of contact points with the surface. When considering the clicking activity of touch devices, the wider spread of contact points results in a higher risk of having users miss the target. Iwase and Murata's [10] experiments also support the conclusion that touch interfaces present a higher rate of errors than mouse pointing interactions.

A common explanation for the inaccuracy of touch devices relates to the fat finger problem [18]. The fat finger problem model considers that the width of the finger has a negative impact on the interaction. As users touch the surface, it is not clear which exact point should be considered as input by the computer, since the finger contact area is wider than that of a pixel-level pointing device such as the mouse. At the same time, the finger or arm used during the touch interaction occlude the target and prevents the target click point from providing visual feedback.

Holz and Baudisch [8] provide another model to describe the inaccuracy of touch devices. Their model of perceived input pointer considers that when a user tries to acquire a target, the actual pointing location considered by the computer is misaligned by a couple of millimeters from the target location where the finger is placed. Moreover, the fact that touch devices report the touch location at an offset increases the risk of missing the target.

In order to overcome the inaccuracy in touch displays, many studies focus on defining ways to design user interfaces which prevent users from missing their targets. Following this line of research, Huang et al. [9] investigated the factors that influenced the usability in LCD touch screens. The study reports that the factor which mostly affects usability according to their experiments is the touch field (size, location, space and density) of icons. Other studies that provide guidance on how to design an application to make it more usable for touch interfaces involve contact area widgets [12] that recommends the use of sliding widgets and contact area activation rather than using a single[1] pixel activation UI; and Apple's iOS Human Interfaces Guidelines[1] that recommend that targets be at least 44x44 pixels in area. It is worth noticing that these guidelines require applications to be designed by developers according to their recommendations. However, when specifically considering the Web, it would require a great amount of effort to re-build and re-deploy all websites and web applications, considering these guidelines and recommendations.

In this context, it becomes necessary to develop new strategies to reduce the number of errors when users are interacting with interfaces that do not follow usability guidelines for touch interfaces. These strategies adapt the interaction and presentation mechanisms of web applications in order to overcome the limitations of

[1] http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/UEBestPractices/UEBestPractices.html

low accuracy for touch displays. One clear example of an interaction adaptation strategy is the zooming based technique [3]. In order to acquire a target, the user can zoom in to enlarge the target until s/he can comfortably touch it, a similar approach to the pinch-to-zoom functionality present in Apple touch screen devices.

Albinsson and Zhai [2] argue that when zoomed to a subarea of the interface, the user loses the global context view of the application. Among techniques that improve accuracy for touch displays while not loosing the global context view of the application, it can be highlighted: Take-Off [16], Cross-lever, Precision- handle [2] and finger prints and position extraction [8].

It is worth noticing that these techniques make touch less direct, what might reduce the intuitiveness of the input (Take-Off, Cross-level and Precision-handle) [8] or require improvements in the hardware of touch devices (finger prints and position extraction).

Our approach, the link-offset-scale technique, tries to keep the direct-input behavior, while making no additional requirements towards the hardware of conventional touch screen devices and improving the accuracy of the interaction. The link-offset-scale functionality works specifically in the web context, using standard web technologies (Ajax and HTML5) to identify the parts of a webpage that require pointing assistance and adapt the webpage presentation in order to minimize the number of link acquirements during the interaction.

3 Adapting Link Presentation - the Link-Offset-Scale Functionality

Clicking in hyperlinks is the most frequent cause of navigation actions inside a web browser [5]. However, considering the growth in popularity of touch screen devices, there is an eager need to develop web solutions that fit the touch interaction scenario, improving the error rate for web application usage in touch devices.

In this context, we present the link-offset-scale technique, to improve the usability of touch interactions in the Web. The goal of the link-offset-scale functionality is to reduce the number of click errors for touch screen devices, by providing feedback information about the links that have been targeted by the user. The mechanism tries to solve the fat-finger problem [18] and the perceived input pointer [8] models by presenting feedback information about links available on webpages, before the user actually activates them.

The feedback information consists of the same HTML content that is placed inside each link of the webpage, but the information is scaled to 3 times the link's actual size (300% of the link's original size). As users touch an specific link (analogue to the mouse down event interaction in web applications) the feedback information about that specific link is presented. The feedback information presentation in the link-offset-scale technique is illustrated in Figure 1. As the feedback information is presented, an animation effect is also applied to the feedback information HTML element, in order to get the user's attention (fade in and movement effect).

Fig. 1. Outline of the link-offset-scale functionality: as the user positions his finger over a specific link (touching the display), the browser presents the content of that link in a separated magnified element next to the user's finger, so as to be in his field of view

The feedback information is placed on top of the target link's original position, with a right offset, if the link is positioned within 50% of the left side of the webpage, or a left offset, if the link is positioned within 50% of the right side of the webpage. The feedback information is positioned on top of the link in order not to have its view blocked by the user's fingers, hands or arms, during the touch interaction. The left or right offset are applied in order to try to avoid having feedback information blocking other links as users move their fingers through links while touching the screen, and considering that many interaction design patterns established for web applications[2] place different links on top of one another (Headerless Menu design pattern, for example) or besides one another (Main Navigation Menu design pattern, for example).

As the users take their fingers off the link (mouse up event), the feedback information is hidden. If the users stop touching the surface while they have their fingers over a link element, the link is activated and the webpage is redirected to the link's URL reference. This behavior is applied to all link elements presented in the webpage. The link-offset-scale mechanisms allow users to check the feedback information before actually acquiring the target link. If the feedback information does not match the users' target clicking link, they can adjust their finger positions in order to activate the correct link. We expect that this possibility of adjustment will reduce the number of errors made by users during the touch interaction.

The click event in a conventional touch device is interpreted as a link activation activity inside a web browser. In the link-offset-scale functionality, we break the click event into three other events: mouse down, mouse over and mouse up events. As the user dispatches a mouse down event, the browser starts listening to mouse over events. If a mouse over event to a link element is dispatched, the feedback information for that link is presented to the user. From there, the user can decide if s/he has targeted the correct link or if finger position adjustments are required. If the user has touched the link s/he meant to, s/he can stop touching the display in order to dispatch a mouse up event and be redirected to the selected URL link. If the user has

[2] http://www.welie.com/patterns/

targeted a wrong link s/he can try readjusting his/her finger to reach the expected link. The comparison between the conventional touch device experience and the link-offset-scale experience is illustrated in Figure 2. In case the users decide to interrupt the activity whilst their fingers are still touching the screen, they can put their fingers over a non-link HTML element (an element that does not provide feedback information), and stop touching the screen. This action will prevent any link activation and page redirection from taking place.

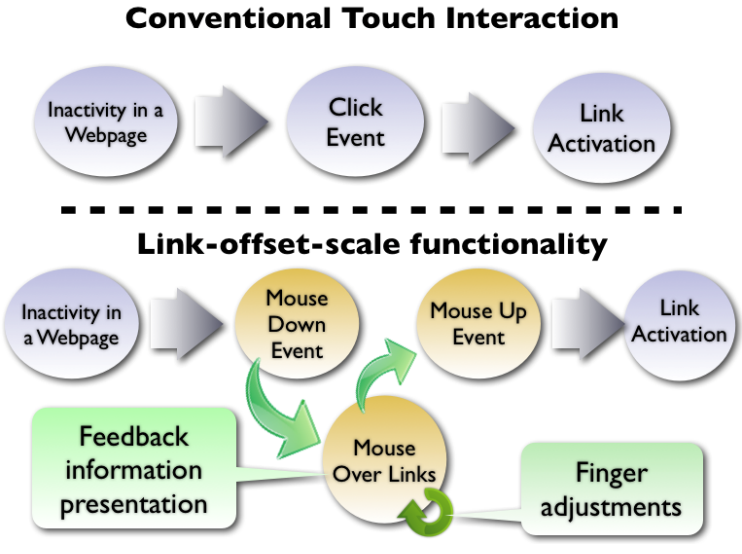

Fig. 2. These state charts illustrate the differences in behavior between the conventional touch experience and the link-offset-scale functionality touch experience

Our mechanism can be interpreted as a hybrid between the Take-Off approach [16] (since we provide targeting aids and break the acquisition task in targeting and then acquiring) and the zooming based technique [3] (as we provide a magnified version of the links information for users). However, it is worth noticing that our approach does not provide an offset pointer to the touch interaction, and thus, the link-offset-scale functionality does not make touching less direct than the Take-Off approach. And the zooming effect does not imply losing contextual information from the general view of the interface. Our mechanism can also be seen as an extension of the *Previewing Information Operation* [13], which proposes generalizing the presentation of a preview of the destination content [15] in order to facilitate the user's decision about when to follow the link, while keeping the original context – in our case, however, the same information presented in the source page is presented to help the user to make this decision.

As a drawback for the proposed approach, we expect the link-offset-scale functionality to increase the average pointing time required to acquire a link element. The possibility of increasing the average pointing time results in the link selection activity's increase in complexity in the link-offset-scale functionality, when compared to the

conventional touch device behavior, as illustrated in Figure 2. However, in the Web context, a click error event does not have any impact in pointing time, but it increases the general task completion time, since the user has to wait for the new webpage to load in order to verify if the pressed link is the meant one, and, in case it is not, press the back button and try to acquire the right link again. These facts require empirical validation to assess how the general task time is affected by the use of the link-offset-scale functionality in order to verify if the increase in pointing time is actually compensated by the decrease in number of errors and loading time of the webpages.

Another action frequently carried out in web applications is scrolling. Web pages usually rely on scrolling to present all their content to users. In conventional touch interaction software, the use of swipe touch actions to scroll the webpage is presented as a well-accepted affordance option in interface design. The use of this type of affordance can be observed in Apple touch devices, Windows 7 drivers for touch screen devices, Google Android smartphones, among others. However, the swipe-to-scroll action prevents the link-offset-scale functionality from being used. If users try to move their fingers over the links in order to get the feedback information, the device will recognize a swipe action in the surface and scroll the screen, instead. So the swipe-to-scroll affordance cannot be applied concurrently to the link-offset-scale mechanism, at least when considering the swipe-to-scroll affordance conventional implementation (that recognizes the swipe gesture anywhere on the surface).

Taking that into account, in order to allow the proposed link-offset-scale functionality to work properly with the swipe-to-scroll affordance, we re-implemented the swipe-to-scroll gesture recognition feature in a way that it works solely on the extreme right side of the screen. More specifically, the swipe-to-scroll affordance will only be activated if the gesture is realized within 10% of the rightmost side of the surface. Any other gesture realized in the other 90% of the surface will work according to the link-offset-scale mechanism. Besides that particular change in the swipe gesture recognition, we also increased the scrolling sensitivity towards the swipe action in order to improve the scrolling time in the web application.

We have implemented a functional prototype for the link-offset-scale mechanism together with our custom implementation of the swipe-to-scroll affordance as a Mozilla Firefox addon, using the Addon-SDK[3] with other standard web technologies (HTML, JavaScript and CSS).

4 Evaluation

Our primary hypothesis is that the link-offset-scale functionality actually reduces the number of errors in acquiring links in a web application. In order to test our hypothesis, we conducted a usability task-oriented experiment that evaluated the number of errors users acquired when interacting with the system. We also monitored the general task time, so that we could analyze how the number of errors and the link-offset-scale mechanism pointing time impacted the task completion time.

The experiment consisted of comparing the number of errors and completion time for a variety of acquiring link tasks, with users interacting with conventional touch

[3] https://jetpack.mozillalabs.com/

screen interaction software and with the implementation of our link-offset-scale functionality.

Next sections describe experiment details about the participants (Section 4.1), the methodology (Section 4.2), and the results (Sections 4.3 and 4.4).

4.1 Participants

For our experiment we had a group of 14 participants. The participants' age ranged from 22 to 36 years old (12 men and 2 women). Test participants consisted of Computer Science graduate students from the ICMC-USP (Institute of Mathematical Sciences and Computer Science at University of Sao Paulo). All participants reported having used touch interfaces at least once a month in bank cash machines, where three of them used it up to twice a week and six used it more than twice a week.

4.2 Experiment Setup and Methodology

We used a Positivo UNION TOUCH 2500[4] as the touch device for the experiment. Even though the Positivo UNION TOUCH 2500 recognizes multi touch gestures, the tasks we carried out consisted only of single touch gestures. The touch-screen devices had Windows 7 installed. When executing a test with the conventional touch software, the computer ran with its standard configurations. When executing a test with the link-offset-scale mechanism, we disabled the standard gesture recognition feature of the operational system in order not to have the swipe-to-scroll operational system interfering with our own implementation of that affordance.

The experiment consisted of comparing the number of errors and completion time for a variety of acquiring link tasks, with users interacting with conventional touch screen interaction software and with our implementation of the link-offset-scale functionality.

Tasks consisted of simple acquiring link activities. The links were placed inside the main content of the webpage or inside UI components detailed as Interaction Design Patterns[5]. The links are also part of real-world websites available on the Web. Some tasks also required the use of scrolling functions in order to evaluate whether the modification of the swipe-to-scroll affordance affected the experiment results. Next, we describe the list of tasks, and the websites and Interaction Design Patterns associated to them:

- **Task 1:** textual element (paragraph HTML element) in http://gc.blog.br.
- **Task 2:** Tag Cloud Interaction design pattern in http://gc.blog.br.
- **Task 3:** Tag Cloud Interaction design pattern in http://gc.blog.br.
- **Task 4:** Headerless Menu Interaction design pattern in http://gc.blog.br.
- **Task 5:** Doormat Menu Interaction design pattern in http://www.wikipedia.org.
- **Task 6:** Accordion Menu Interaction design pattern in http://www.wikipedia.org.

[4] http://www.positivoinformatica.com.br/www/pessoal/tudo-em-um/tudo-em- um/positivo-union-touch-2500/visao-geral

[5] http://www.welie.com/patterns/

- **Task 7:** Directory Navigation Interaction design pattern in http://www.wikipedia.org.
- **Task 8:** Footer Site Navigation Interaction design pattern in http://www.wikipedia.org.
- **Task 9:** Faceted Navigation Interaction design pattern in http://www.icmc.usp.br.
- **Task 10:** Fly-out Menu Interaction design pattern in http://www.icmc.usp.br.
- **Task 11:** Fly-out Menu Interaction design pattern in http://www.icmc.usp.br.

It is worth noticing that previous knowledge about one of the websites used in the study (http://gc.blog.br, http://www.wikipedia.org or http://www.icmc.usp.br) could lead a participant to complete the task faster and thus generate a biased result of the experiment. Bearing that in mind, we pointed out and highlighted the target link for the participants before each task. Thus, the participants' task completion time was not affected by their familiarity with the task's web application and they were not required to look for the link within the webpage.

If a participant missed acquiring a link of the task, the system presented an error message and the participant was asked whether s/he would like to continue with the experiment or give up the task and go to the next one. It is worth noticing that the error message was instantly presented to users after any misplaced acquiring interaction. In real world tasks, however, there would be no error message and users would need to wait the next page to load to, then, identify whether their activity led to an unexpected result. Therefore, errors in the experiment had less impact on the task time than there would be in real world clicking tasks.

It is also worth noticing that errors were interpreted differently according to the system being tested. While testing the conventional touch interaction, all misplaced click events were classified as errors in the experiment. While testing the link-offset-scale functionality, all mouse up events that happened over an HTML link element were classified as errors, however mouse up events dispatched over non-link HTML elements were not classified as errors (standard action for canceling a link acquirement in the link-offset-scale mechanism).

The general procedure undergone by all participants consisted of the following steps:

1. Base explanation about the test goals of measuring the number of errors and task completion time.
2. Conventional touch mechanisms test session:
   a. Base explanation about how to use the conventional touch interaction mechanisms in webpage navigation.
   b. Start of the test session where the participant was asked to complete the 11 tasks using conventional touch mechanisms.
3. Link-offset-scale mechanisms test session:
   a. Base explanation about how to use the link-offset-scale interaction mechanisms in webpage navigation.
   b. Start of the test session where the participant was asked to complete the 11 tasks using the link-offset-scale touch mechanisms.

As it is possible to notice from these steps, all participants interacted with both touch interaction mechanisms (conventional touch functionality and the link-offset-scale functionality). However, the order they used each of the touch interaction mechanisms could generate biased experiment results. Using one mechanism could lead users to misinterpret the other. Therefore, we randomly changed the order of step 2 (Conventional touch mechanisms test session) and step 3 (Link-offset-scale mechanisms test session), so that half the participants executed step 2 before step 3, and vice versa.

In the experiment, the participants were not given time to become familiarized with the link-offset-scale functionality before starting the evaluation session. The main goal of the test was to verify if our approach reduced the number of errors in touch screen displays in web applications. If participants were allowed to familiarize themselves with the link-offset-scale mechanisms, the evaluation would be impacted in favor of the link-offset-scale approach, since most users were familiar with conventional touch interaction mechanisms and had never used the link-offset-scale mechanisms before.

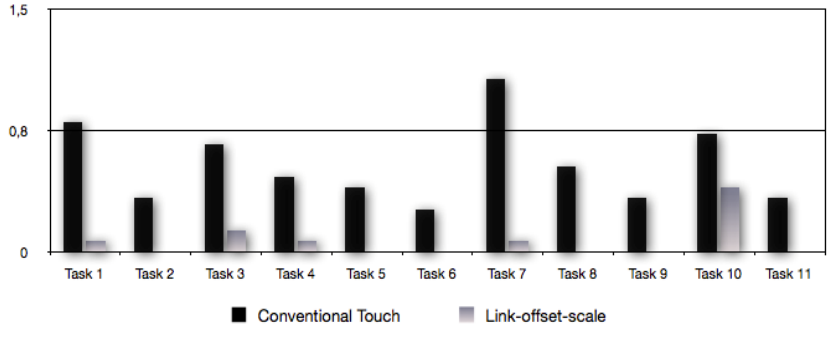

Fig. 3. Average number of errors for each task, according with the use of conventional touch screen presentation mechanisms and the enhanced presentation mechanisms with the link-offset-scale functionality

After having completed both test sessions, we asked the participants for comments and feedback on their experience with both touch interaction mechanisms.

4.3 Error Rate Analysis

In the experiment, it was verified that the number of errors was lower for all tasks when users were interacting with the link-offset-scale functionality. Figure 3 illustrates the average number of errors per task. In this figure, it can be observed that the average number of errors was lower for all tasks in our sample.

Tasks 1, 2, 4, 7, 8, 9 and 11 presented significantly less errors with the link-offset-scale mechanisms when compared to the conventional touch mechanisms, using a paired t-test analysis (with p-value inferior to 0.05). The average standard deviances and p-values data for all tasks are showed on Table 1.

Table 1. Average standard deviation for the number of errors for the conventional touch mechanism (CM as conventional touch average and CD as conventional touch standard deviation) and link-offset-scale interaction mechanisms (LM as link-offset-scale average and LD as link-offset-scale standard deviation) and the t-test p-value for analyzing whether the link-offset-scale mechanism presents less errors than the conventional one

Tasks	CM	CD	LM	LD	p-value
Task 1	0.8	1.16	0.06	0.26	0.01764
Task 2	0.33	0.74	0	0	0.04806
Task 3	0.66	1.13	0.13	0.53	0.06762
Task 4	0.46	0.65	0.06	0.26	0.02685
Task 5	0.4	1.15	0	0	0.0947
Task 6	0.26	0.61	0	0	0.05193
Task 7	1.06	1.16	0.06	0.26	0.002753
Task 8	0.53	0.93	0	0	0.02005
Task 9	0.33	0.49	0	0	0.009318
Task 10	0.73	1.12	0.4	0.85	0.1678
Task 11	0.33	1.08	0	0	0.1193

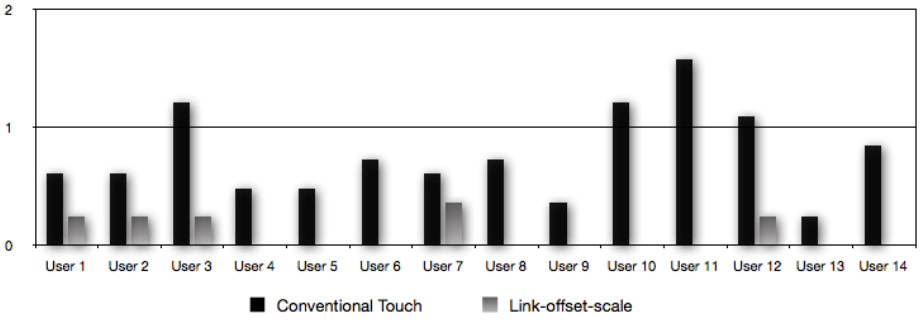

Fig. 4. Average number of errors per user, according to the use of conventional touch screen presentation mechanisms and the enhanced presentation mechanisms with the link-offset-scale functionality

Considering the average number of errors per user, we could also verify some positive results. All users performed better with the link-offset-scale mechanisms than with the conventional touch ones. Figure 4 presents the average number of errors per user.

4.4 Task Time Analysis

The experiment's results showed that, in the sample, the link-offset-scale functionality had a longer average task completion time for each task in comparison to the conventional touch interaction mechanisms. The only exception was for Task 7, which presented an inferior average completion time when users were interacting with the link-offset-scale mechanisms. The explanation for Task 7's different behavior is that it presented the highest difference in average number of errors when comparing the two approaches (1.066 average number of errors when interacting with the

conventional touch interaction mechanisms against 0.066 when interacting with the link-offset-scale functionality). Errors during the test session increased the task completion time, since users were required to identify the error messages on the screen and try again. In Task 7, the increase in task completion time due to the number of errors during the interaction with the conventional touch interaction mechanisms compensated for the longer pointing time required by the link-offset-scale functionality.

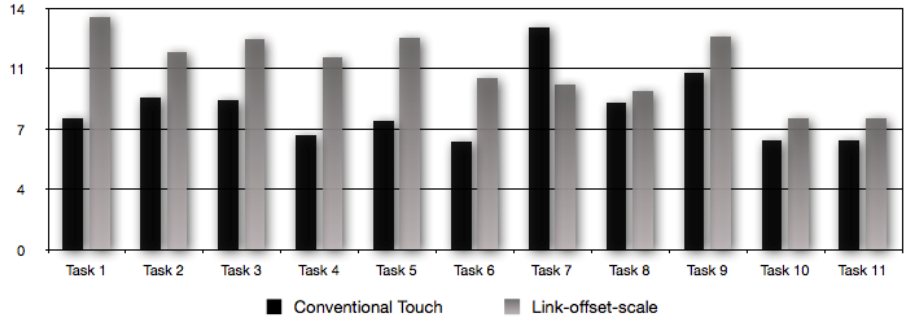

Fig. 5. Average time required for completing each task according to the use of conventional touch screen presentation mechanisms and the enhanced presentation mechanisms with the link-offset-scale functionality

It is important to notice that only two tasks presented significant differences between the task completion time for the conventional touch interaction mechanisms and the link-offset-scale mechanisms (Tasks 1 and 4), in a paired t-test with p-value under 0.05. Details about the t-test performed for each task are presented in Table 2.

Table 2. Average standard deviation for the task completing time for the conventional touch mechanisms (CM as conventional touch average and CD as conventional touch standard deviation) and link-offset-scale interaction mechanisms (LM as link-offset-scale average and LD as link-offset-scale standard deviation) and the t-test p-value for analyzing whether the link-offset-scale mechanism presents a greater task completing time than the conventional touch mechanism

Tasks	CM	CD	LM	LD	p-value
Task 1	7.6	4.12	13.53	11.40	0.03690
Task 2	8.86	6.94	11.46	6.79	0.09581
Task 3	8.86	6.13	12.2	15.21	0.2162
Task 4	6.66	4.81	11.13	5.22	0.0006582
Task 5	7.46	11.66	12.33	17.95	0.2009
Task 6	6.26	5.52	9.93	9.30	0.09794
Task 7	12.93	7.55	9.6	7.79	0.8727
Task 8	8.53	6.76	9.2	4.53	0.3786
Task 9	4.06	2.23	6.66	8.46	0.1087
Task 10	10.26	7.32	12.4	9.36	0.2588
Task 11	6.33	5.38	7.66	5.40	0.2458

Considering the average task completion time for each user, it can be verified that 9 users performed better with the conventional touch interaction mechanisms, while 5 users performed faster with the link-offset-scale mechanism. These data are illustrated in Figure 6.

5 Discussion

All subjects presented greater security (fewer errors) when interacting with the link-offset-scale functionality, as detailed in Figure 4. It was also verified that the average number of errors, depending on the task, support our main hypothesis that the link-offset-functionality reduces the number of misplaced link acquirements during the interaction, considering the test results in the sample. The feedback information presentation before the actual link selection action results in a better performance in terms of number of errors, whilst still keeping the direct input nature of touch devices.

The experiment also verified that the average task completion time is increased when users interact with the link-offset-scale functionality. That is a result of the increased pointing time required when interacting with the link-offset-scale mechanisms caused by the greater interaction complexity illustrated in Figure 2. However, the task completion time increment due to errors in the experiment is inferior to the task completion time increment observed in real world examples. Since in real-world examples errors require users to wait for the loading time of the website, recognizing that the website was wrongly selected, go back to the previous website and then restart the interaction. Further evaluation needs to be carried out in order to completely validate whether the general task completion time is longer when interacting with the link-offset-scale mechanism in the Web.

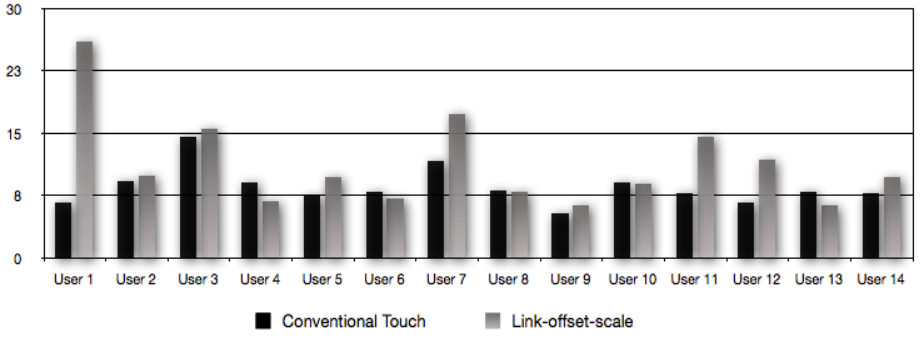

Fig. 6. Average time required for completing tasks for each user, according to the use of conventional touch screen presentation mechanisms and the enhanced presentation mechanisms with the link-offset-scale functionality

The link-offset-scale mechanism addresses the problem of low accuracy in touch screen displays, and its results are similar to other approaches (Take-Off, Cross-lever and Precision Handle) presented in the Related Work section (Section 2): there is a lower number of errors in the approach - however, a higher pointing time can be observed when interacting with the system. Nevertheless, the link-offset-scale

approach maintains the direct input behavior of touch screen displays (different from the Take-Off, Cross-lever and Precision Handle strategies) and requires no additional hardware improvement (finger prints and position extraction) in order to reduce the number of errors in the touch interaction.

At the end of each testing session, we asked the participants to give us feedback on their interaction experience with the conventional touch interaction mechanisms and the link-offset-scale functionality. From the participants comments we have identified the following advantages and disadvantages that were not formally measured nor defined in the quantitative results section:

- **Differences in performance among tasks:** The high error rate in Task 1 was probably a result of it being the first task in the session - participants were more likely to make errors. Task 7 presented the highest error rate in the conventional touch mechanisms evaluation sessions as a result of the size of the target link of the task (located in the footer of the webpage). As the link-offset-scale mechanism provides a zoomed version of the text information placed inside the acquired link, the size of the link did not severely impact the rate of number of errors. Finally, task 10 and 11 consisted of a fly-out menu interaction design pattern and, therefore, required a more complex interaction in order to acquire the links placed in them. The fly-out menu requires the user to place the cursor (touch) in some parts of the menu. As the user places the cursor in these parts, additional parts of the menu are shown, and the user is able to click/acquire the target links in the task. We believe that the more complex nature of the interaction in fly-out menus resulted in a more severe impact on task 10, since it was the first task realized with this type of User Interface component.
- **Slow processing for the feedback information presentations:** some users stated that the feedback information for the links were not synchronous to the movement of their fingers over links. The feedback information was presented with an animation effect, in order to catch the users attention. However, the general functionality was implemented using JavaScript in the browser, therefore the processor was possibly overwhelmed by the number of animation effect executed with an interpreted programming language. In this context, we will consider removing the animation effect or reduce the animation time, so that feedback information is presented synchronously to the users fingers movement in the screen. It is also worth noticing that we run our experiments using a Mozilla Firefox Browser version 3.6.12, that does not contain hardware acceleration for visual effects in web applications. And hardware acceleration enabled browsers might reduce the slow observed by users while processing for the links feedback information presentation routines.
- **Advantages in interacting with smaller links:** most users reported that they were greatly benefited by the link-offset-scale functionality when trying to click on smaller links. These users reported that the link-offset-scale functionality is not required for all links, but should be applied for smaller links and widgets with links close to one another. Another advantage observed was that the magnifying functionality of the link-offset-scale mechanisms at a scale of 300% helped users to read and identify link feedback information.

- **Link-offset-scale selection strategy:** during the experiments, it was observed that users normally touched a non-link element of the screen and then dragged their finger to the target link they were aiming at, when interacting with the link-offset-scale mechanisms. This link selection strategy might have been used because, despite the feedback information presentation, the users were not aware of how to adjust their fingers when they were positioned over the wrong link. This was due to the fingers or hand blocking the links close to the one the user had his/her fingers on. Therefore, users started the mouse down event from a non-link element in order to draw a linear route that intercepted the target link and then they dragged their fingers over this linear route until the feedback information presented matched the one that described the target link. This behavior was probably one of the causes for the superior pointing time required by the link-offset-scale functionality, and solving it might improve the task completion time for the mechanism.
- **Lateral scroll difficulties:** it was observed that some users presented difficulties when interacting with the custom implementation of the swipe-to-scroll affordance, such as when clicking links that appeared under the swipe-to-scroll activation area. In order to activate links under the scroll area, the user would have to start his/her drag movement from outside the scroll area, then enter the scroll area whilst still touching the screen, and then remove his/her fingers off the surface when the target link feedback information was presented. This interaction pattern characterizes a non-intuitive activity and should be improved in next versions of the link-offset-scale functionality. Some users also reported that they lacked feedback information about when the custom implementation of the swipe-to-scroll functionality was active.
- **Scroll speed:** our custom implementation of the swipe-to-scroll affordance had an increased sensitivity, when compared to the standard implementation of the affordance. A few users commented that the increased sensitivity made scrolling activities faster, which might generate biased results for the task completion time data analysis. However, the scrolling did not affect all tasks and the scroll operation was only executed once during the task, regardless of the number of errors captured during the interaction. Therefore, we believe that the scrolling increased sensitivity did not generate biased results for the task completion time analysis.

It is worth noticing that all participants were computer experts (all graduate Computer Science students), where half of them can be classified as touch interaction experts, given the high frequency with which they use touch devices. In this context, it becomes difficult to evaluate the learning complexity of the proposed approach. As a future work, it would be interesting to run the experiment with a different user group, considering specifically computer inexperienced users, to verify whether the results and conclusions present different tendencies and evaluate whether the interaction complexity generated by the link-offset-scale functionality also impacts the usability of the touch interaction.

6 Final Remarks

This work aimed at presenting the link-offset-scale touch interaction mechanism for improving the usability for touch screen devices. The link-offset-scale mechanism makes use of web application structure meta-data (identifying links) to provide feedback information about the selection of links in touch interfaces. This feedback information is presented as the users move their fingers over the surface, while touching HTML link elements available on a webpage. As the feedback information is presented prior to the actual link activation, users are able to adjust their link selection in order to avoid errors during the interaction. The link-offset-scale mechanism's primary goal is to reduce the number of errors that users have while interacting with touch screen devices in the Web.

We conducted a usability experiment in order to evaluate whether the proposed approach reduces the number of errors during an interaction. The experiment results show that the number of errors was significantly lower when interacting with the link-offset-scale functionality. However, it was observed that the general task completion time was higher for the interactions that used the functionality.

As future works, we consider re-running the experiment with different user groups (computer inexperienced groups, more specifically), improving the implemented features in order to address the users' concerns about the slow processing feedback information presentation and scroll difficulties. We also plan on deploying multi-platform versions of the functionality in order to compare the users' usability performance when interacting with different touch devices.

Acknowledgments. The authors would like to thank CAPES, CNPq and FAPESP for the financial support.

References

1. Abowd, G.D., Mynatt, E.D., Rodden, T.: The human experience. IEEE Pervasive Computing 1, 48–57 (2002), IEEE Educational Activities Department, http://portal.acm.org/citation.cfm?id=612822.612832
2. Albinsson, P.A., Zhai, S.: High precision touch screen interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2003, pp. 105–112. ACM, New York (2003), http://doi.acm.org/10.1145/642611.642631
3. Bederson, B.B., Hollun, J.D.: Pad++: A zooming graphical interface for exploring alternate interface physics. In: Proceedings of User Interface and Software Technology, pp. 17–26. ACM, New York (1994), http://doi.acm.org/10.1145/192426.192435
4. Berners-Lee, T.: Information management: A proposal. W3C archive (1990), http://www.nic.funet.fi/index/FUNET/history/internet/w3c/proposal.html
5. Catledge, L.D., Pitkow, J.E.: Characterizing browsing strategies in the World-Wide Web. Comput. Netw. ISDN Syst. 27, 1065–1073 (1995), http://portal.acm.org/citation.cfm?id=206477.206540
6. Garret, J.J.: Ajax: A new approach to web applications (2005), http://www.adaptivepath.com/ideas/essays/archives/000385.php
7. Hickson, I.: A vocabulary and associated APIs for HTML and XHTML. W3C Working Draft (October 2010), http://www.w3.org/TR/html5/

8. Holz, C., Baudisch, P.: The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, pp. 581–590. ACM, New York (2010), http://doi.acm.org/10.1145/1753326.1753413
9. Huang, H., Tsai, W.C., Lai, H.H.: Factors influencing the usability of icons in the LCD touch screens. In: Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction: Ambient Interaction, UAHCI 2007, pp. 878–887. Springer, Heidelberg (2007), http://portal.acm.org/citation.cfm?id=1763296.1763396
10. Iwase, H., Murata, A.: Empirical study on improvement of usability for touchpanel for elderly - comparison of usability between touch-panel and mouse. In: IEEE International Conference on Systems, Man and Cybernetics 2002, pp. 252–257. IEEE Computer Society, Los Alamitos (2002)
11. Malacria, S., Lecolinet, E., Guiard, Y.: Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the cyclostar approach. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, pp. 2615–2624. ACM, New York (2010), http://doi.acm.org/10.1145/1753326.1753724
12. Moscovich, T.: Contact area interaction with sliding widgets. In: Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, UIST 2009, pp. 13–22. ACM, New York (2009), http://doi.acm.org/10.1145/1622176.1622181
13. Pimentel, M.d.G.C.: Evaluation of alternative operations for browsing hypertext. In: Conference on People and Computers IX, pp. 145–162. Cambridge University Press, Cambridge (1994), http://portal.acm.org/citation.cfm?id=211382.211395
14. Shneiderman, B.: Designing the user interface: strategies for effective human-computer interaction, p. 466. Addison-Wesley Longman Publishing Co., Inc., Boston (1986)
15. Shneiderman, B.: User interface design for the hyperties electronic encyclopedia (panel session). In: Proceedings of the ACM Conference on Hypertext 1987, pp. 189–194. ACM, New York (1987), http://doi.acm.org/10.1145/317426.317441
16. Shneidermnan, B.: Touch screens now offer compelling uses. IEEE Softw. 8, 93–94 107 (1991), http://dx.doi.org/10.1109/52.73754
17. Tsai, W.C., Lee, C.F.: A study on the icon feedback types of small touch screen for the elderly. In: Stephanidis, C. (ed.) UAHCI 2009. LNCS, vol. 5615, pp. 422–431. Springer, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-02710-9_46
18. Vogel, D., Baudisch, P.: Shift: a technique for operating pen-based interfaces using touch. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2007, pp. 657–666. ACM, New York (2007), http://doi.acm.org/10.1145/1240624.1240727
19. Watanabe, W.M., Neto, D.F., Bittar, T.J., Fortes, R.P.M.: WCAG conformance approach based on model-driven development and WebML. In: Proceedings of the 28th ACM International Conference on Design of Communication, SIGDOC 2010, pp. 167–174. ACM, New York (2010), http://doi.acm.org/10.1145/1878450.1878479
20. Wu, M., Shen, C., Ryall, K., Forlines, C., Balakrishnan, R.: Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems, pp. 185–192. IEEE Computer Society, Washington, DC, USA (2006), http://portal.acm.org/citation.cfm?id=1109723.1110635