

# W3Touch: Metrics-based Web Page Adaptation for Touch

Michael Nebeling, Maximilian Speicher and Moira C. Norrie  
Institute of Information Systems, ETH Zurich  
8092 Zurich, Switzerland  
{nebeling,norrie}@inf.ethz.ch, maximilianspeicher@gmx.de

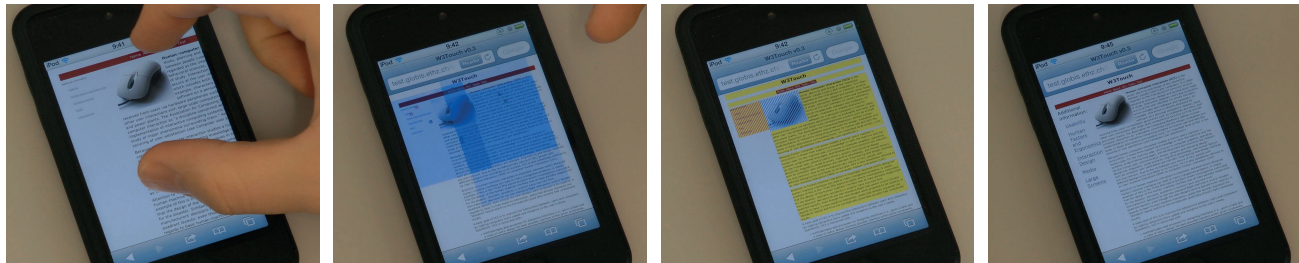


Figure 1: Metrics-based adaptation process using W3Touch: (1) logging, (2) inspection, (3) segmentation and (4) adaptation

## ABSTRACT

Web designers currently face the increased proliferation and diversity of new touch devices which pose major challenges to the design task. This paper presents *W3Touch*—an interface instrumentation toolkit for web designers to collect user performance data for different device characteristics in order to help them identify potential design problems for touch interaction. Web designers can visualise the data aggregated by *W3Touch* and use simple metrics to automate the adaptation process for many different viewing and interaction contexts. In a series of experiments with web designers and users, we show that *W3Touch* is able to detect interaction problems that are hard to find using conventional methods and demonstrate how the tool was successfully used to automate the desktop-to-mobile migration of Wikipedia as an example.

## Author Keywords

Adaptive Interfaces; Touch Interaction; Usability Metrics.

## ACM Classification Keywords

H.5.2 User Interfaces: Input devices and strategies; Screen design

## INTRODUCTION

Web designers currently have to deal with the increased proliferation of primarily touch-operated mobile devices often used for web browsing. They therefore need to think about

how to modify and optimise the design of their sites specifically for touch. Existing browser support alone is often insufficient as it is not only fairly limited, but also varies considerably between devices. The main approach to start from a thumbnail of the page scaled to the width of the device causes readability problems due to the typical fixed-width layout of many sites designed for desktop settings. Moreover, converting hyperlinks and controls such as buttons, list views and text input elements into touch-sensitive areas is not sufficient without also adjusting the padding and spacing. To mitigate the issues, mobile browsers allow users to zoom into areas of a web page as required, but zooming requires user intervention and ultimately detracts from the overall usability.

There are mainly two types of approaches for the adaptation of interfaces to mobile devices. The first is based on interface transformation or generation, typically focusing on web page summarisation [6, 20] or segmentation methods [10, 15] to automatically deal with smaller amounts of screen real estate. The second instead aims to extend the interaction techniques based on such paradigms as collapse-to-zoom [2], proxy-based interaction [18] and multi-point selection [3] for increased precision of touch input. However, this means that touch-related usability issues are not directly addressed by existing methods and are usually treated as a matter of interaction technique rather than design problem.

This paper presents *W3Touch*—an interface instrumentation toolkit that can be configured by web designers to produce analytics of the user interaction in order to help them detect and locate potential design problems for touch. We build on existing techniques for automated usability evaluation [1, 17] and extend them with touch-specific metrics and input tracking methods: (1) to monitor multi-touch and gesture-based interactions often used when scrolling and zooming web content on touch devices, (2) to detect mis-clicks and actions that

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2013, April 27–May 2, 2013, Paris, France.

Copyright 2013 ACM 978-1-4503-1899-0/13/04...\$15.00.

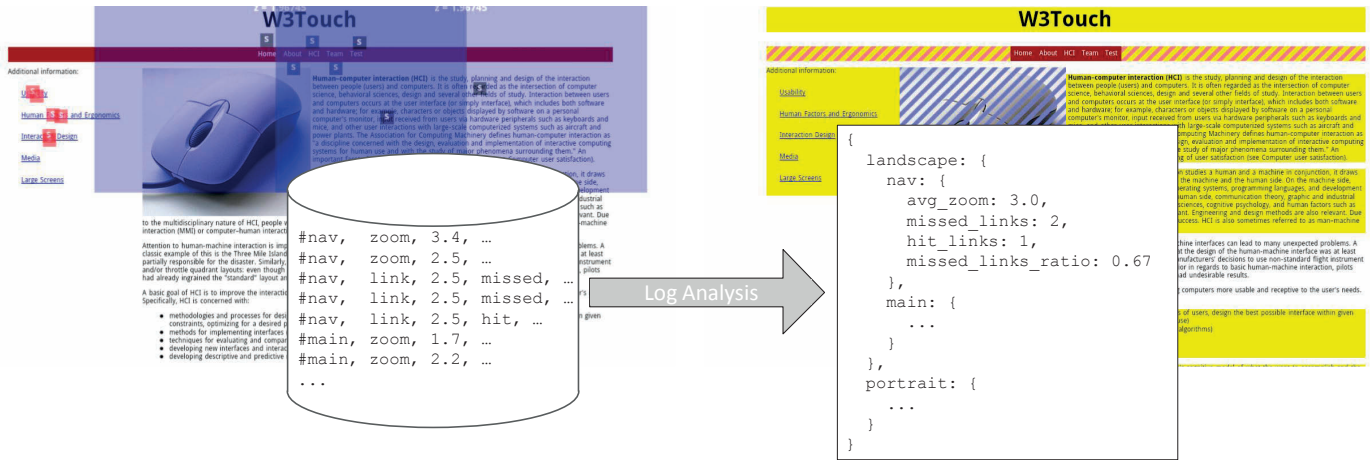


Figure 2: W3Touch instruments the web interface to collect interaction data for each component (zoomed into #nav component, missed or hit link in #nav, etc.) and measure user performance based on different metrics (avg\_zoom, missed\_links, etc.)

were not triggered although intended, and (3) to handle context switches between portrait and landscape mode.

W3Touch contributes a new interface adaptation method supporting automation based on usability metrics and the evidence of interaction problems on different forms of touch devices. Our approach is in contrast to the current trend of manually designing for different touch devices, which requires significant technical knowledge of different software technologies and development methods [9]. W3Touch provides components for aggregating and visualising user interaction data and can be configured with different metrics and rules to automate the adaptation process. With our technique, designers can develop dynamic touch optimisations that let the interface self-adjust according to user and device requirements.

Figure 1 illustrates the adaptation process based on W3Touch for an example page. The adaptation using W3Touch consists of three steps. First, the user interaction is monitored and relevant data collected for each metric. Second, W3Touch implements visualisation techniques for inspection of the raw data and segmentation of the interface into critical components based on thresholds defined for each metric. Based on these visualisations, designers may adjust the thresholds and experiment with different adaptations. Finally, they can deploy an adaptation catalogue fixing the design problems identified for different contexts. In the example, the links in the sidebar navigation are displayed larger and with more spacing, enabling precise selection without the need for zooming.

The next section presents the key ideas behind W3Touch and our first implementation. The evaluation following this focuses on the Wikipedia site which provides different interfaces for desktop and mobile settings. We use a three-step method based on interviews with web designers, an online user study with W3Touch and a follow-up lab study to assess the tool support and the quality of the baseline interfaces compared to a version using W3Touch. The results suggest that W3Touch is able to quantify touch-related usability problems and successfully automate the adaptation to many different touch devices. The paper compares our solution to existing techniques and outlines possible extensions to W3Touch.

## W3TOUCH

As illustrated in Figure 2, W3Touch can be used to instrument existing web interfaces and record interaction data for each use context. Web designers can make use of this information to identify potential design problems and define the adaptation rules necessary to mitigate any issues. To make this task easier for designers, W3Touch provides three major features:

- **Interaction Tracking** W3Touch automatically injects client-side event handlers in JavaScript to monitor touch events that activate links or modify the viewport.
- **Metrics** W3Touch can be configured with different metrics to aggregate and visualise the recorded touch data.
- **Adaptation Catalogue** W3Touch triggers the adaptation rules matching the current device and viewing condition.

At the core of W3Touch are the metrics that are used both for analysing any collected data and triggering adaptation rules based on customisable thresholds. Based on the well-known touch input problem of pointing accuracy [16], we built two default metrics into W3Touch:

- *Missed links ratio* The mis-clicks metric keeps track of how often touches miss an intended target to register actions that were intended, but did not get triggered.
- *Zoom level* The zoom factor metric considers how much users on average need to zoom into different components of the web interface.

High values for these metrics may for example suggest that the size of the target, the spacing between elements, or both, need to be adjusted. Based on critical thresholds, W3Touch can automatically detect and suggest web page components that may need to be adapted for specific settings. Moreover, the zoom levels applied by users could indicate the preferred size and resolution of text and media.

In the following, we present each feature in more detail.

### Interaction Tracking

W3Touch’s techniques for user input tracking build upon the principles described in previous work [1]. However, since

existing event capturing methods usually operate at the very low level of tracking single mouse and keyboard events, we extend them in three ways to provide support for capturing touch input at a higher level of abstraction.

First, W3Touch keeps track of the mobile browser viewport to enable the detection of pinch-to-zoom gestures, double-tap auto-focus actions, scrolling events and changes of the device orientation. These actions are usually composed of several browser events and therefore difficult to track with existing low-level event tracking mechanisms. Every time the window is scrolled or zoomed, the viewport is captured in terms of the current scrolling offset and the dimensions of the window. The zoom level is then determined by dividing the actual fixed width of the HTML document by the current viewport width. Visible elements are identified by treating the viewport like another element in the interface and building the intersections between DOM elements and the viewport.

Second, in addition to logging successful taps on interactive elements such as hyperlinks or input fields, W3Touch also registers touches that potentially missed the intended target. This is done automatically by augmenting the web interface with touch-sensitive, but invisible, areas of 0.4cm range around all interactive elements to register touches nearby by the target. This threshold was determined in several studies by other researchers [16] and also showed good results in our tests. Figure 3 shows the enhanced interface of the CHI 2013 web site with the extended tracking areas highlighted in red. Mis-clicks are automatically captured each time a touch event is triggered on the red area surrounding the interactive element rather than the respective element's touch area itself.

Third, W3Touch includes mechanisms for reducing the amount of tracking data. For example, a series of touch events registered as part of a gesture or continuous scrolls in the same direction are recorded as a single interaction.

All event data is enhanced with context data in terms of the device type, viewport size and orientation. The collected data is buffered and sent to the server at suitable intervals to keep the web interface responsive, unless a link to a new page is activated, in which case all data is sent immediately before leaving the current page. The tracking data is processed on the server-side using common log analysis techniques.

### Metrics

To do the analysis, the data for the different metrics is first grouped by element and aggregated per context as illustrated in Figure 2, and then processed, e.g. to determine the average zoom level `avg_zoom` (i.e. zoom factors averaged across similar devices), the number of mis-clicks vs. successful hits and the resulting `missed_links_ratio` metric (i.e. number of mis-clicks divided by number of all clicks).

Based on this information and configurable thresholds, W3Touch automatically segments the web interface into critical components. As a first simple approach, W3Touch defines critical components as all active DOM elements whose missed links ratio is above a threshold of 0.3 as well as DOM elements and text nodes that were often zoomed beyond a customisable level of 1.5, if they cover the majority



Figure 3: W3Touch extends touch areas to register mis-clicks

of the viewport. These default thresholds were determined in smaller experiments with good results for several sites. W3Touch then either marks each DOM element directly if they are associated with an ID. If a critical element cannot be referenced by ID, W3Touch finds the closest parent element with ID whose bounding box contains it.

The motivation behind this ID-based approach is that wrapper elements, and therefore often the web page components that drive the structure, are typically labelled for dynamic scripting or formatting based on CSS design rules. As a result of this, they can be easily accessed when traversing the DOM tree, which provides a fairly robust way to uniquely identify the elements or the container element in need of adaptation. Note that the segmentation must therefore not necessarily correspond to the level of each critical element, and the proposed technique also provides a way for developers to fine-tune or generalise the segmentation process by using more or fewer IDs. We are aware of the limitations of this technique when it comes to changes of the internal structure of pages, and more advanced DOM structure-based similarity measures [4] could be added to the approach. However, in our experiments, the ID-based segmentation approach worked fairly well also to reuse the collected data between different versions of the same page, or even different pages within the same site, as long as the internal structure does not change significantly.

For the tracking data, W3Touch provides two different types of visualisations shown in Figure 2 to provide visual feedback to designers and support them in designing new adaptations. The first mode displays the raw tracking data before segmentation using coloured semi-transparent overlays for the touch



area around mis-clicks as well as zoomed-in areas of the exact position and dimensions of the viewport in terms of the entire web interface. In the second visualisation, W3Touch highlights the critical elements that were automatically identified based on the aforementioned segmentation step and configured thresholds for the tracking data.

### Adaptation Catalogue

Based on the aggregated tracking data, it is then possible to define adaptation rules in W3Touch's adaptation catalogue. Such adaptation rules can be defined based on the current tracking data and may be restricted to specific devices and viewing conditions only. Note in the example below how real interaction data collected by W3Touch is used both as a threshold and a parameter to obtain the adaptation rule.

```
if (W3Touch.isSmartphone() &&
    W3Touch('#nav').missed_links >= 0.5) {
    $('#nav a').css('font-size',
        W3Touch('#nav').avg_zoom*100 + '%');
}
```

The above rule scales the size of text links in an example navigation component `#nav` according to the average zoom level measured for this component on smartphones, if the missed links ratio exceeds the specified threshold. Since the adaptation is defined relative to the current font size, other rules will remain intact and could change additional design aspects.

This adaptation rule for the `#nav` component is just an example of a rule that could apply to all web interfaces with sidebar navigation. Since W3Touch is based on web standards, rules can target and adapt any elements of the rendered web interface as long as they are fully represented in the HTML DOM (notable exceptions are Flash, Silverlight and custom HTML5 canvas implementations, where only the container element that embeds the content is accessible). The rest of the paper will focus on using W3Touch for adapting text-centric web sites such as Wikipedia. However, we have explored the techniques implemented in W3Touch for other sites and different kinds of web applications in several projects [23]. For example, we used W3Touch on image-heavy contents to determine the preferred viewing size depending on the context and therefore appropriate resolutions and compression quality of media objects on different types of mobile devices.

### Implementation

W3Touch is built using jQuery for the client and PHP with MySQL on the server. Existing web sites can be enhanced with the tracking tools simply by linking our implementation as external JavaScript and providing a configuration for the server-side components. By default, interaction tracking is enabled for all web page elements with ID, but if certain elements are to be included or excluded, then this can be specified using jQuery selectors<sup>1</sup>.

Our first proof-of-concept of W3Touch was implemented and fully tested for WebKit-based browsers<sup>2</sup> available on all popular mobile devices. However, as W3Touch builds on frame-

works such as jQMultiTouch [25], it is designed to work on many different touch devices and implements techniques for eliminating cross-browser compatibility issues as well as resolving differences between several smartphone and tablet models, where for example rotation is sometimes handled differently. In addition, W3Touch uses MobileESP<sup>3</sup> for device detection as this is not always possible from the browser information and screen dimensions alone because some smartphones have higher resolutions than tablets.

Using asynchronous HTTP post requests and the various optimisation techniques for reducing the client-side tracking generally help to keep the interface responsive. However, our experiments showed that W3Touch does not perform well on the older generation of touch devices where the interaction seemed comparatively slow. The performance also drops with large DOM structures or where many small, nested page elements are included in the tracking. However, this can be compensated by configuring beforehand which elements are tracked using jQuery selectors.

When loading a W3Touch-enhanced page, the system fetches the metrics for the user's browsing context and applies the adaptation catalogue. As in the previous example, adaptations are defined using a combination of JavaScript and CSS instructions that will be fetched from the server-side and added dynamically to the head element of the DOM. As a result, the adaptations are downloaded onto the client and automatically executed by the browser. When the device orientation changes, this is considered a different use context for which the current adaptations are undone and the catalogue reapplied with values for the new orientation.

More information and the source code are available online.<sup>4</sup>

### EVALUATION

We conducted an evaluation of W3Touch with web designers and users in order to test how well W3Touch can support the adaptation to mobile touch devices and how much this process can be automated based on the integrated metrics. Our investigation focused on Wikipedia as an example that is interesting because it already comes in two versions (Figure 4). The default version is optimised for desktop screens using a typical component-oriented layout with a horizontal navigation in the header and vertical sidebar navigations left and right as well as the main column for content filling most of the screen. On the other hand, Wikipedia Mobile is specifically designed for small-screen devices as it integrates the components into a single-column layout, but removes the left sidebar navigation, and also makes smaller optimisations for touch concerning font size and spacing between links. The evaluation was conducted in three steps.

In the first step, we conducted interviews with six web designers on the issues they could see when interacting with the original Wikipedia on two touch devices of different sizes, and how they would address them. This part of the evaluation allowed us to compile a list of potential issues for touch. We also obtained a number of possible adaptations that we

<sup>1</sup><http://api.jquery.com/category/selectors>

<sup>2</sup><http://www.webkit.org>

<sup>3</sup><http://blog.mobileesp.com>

<sup>4</sup><http://dev.globis.ethz.ch/w3touch>

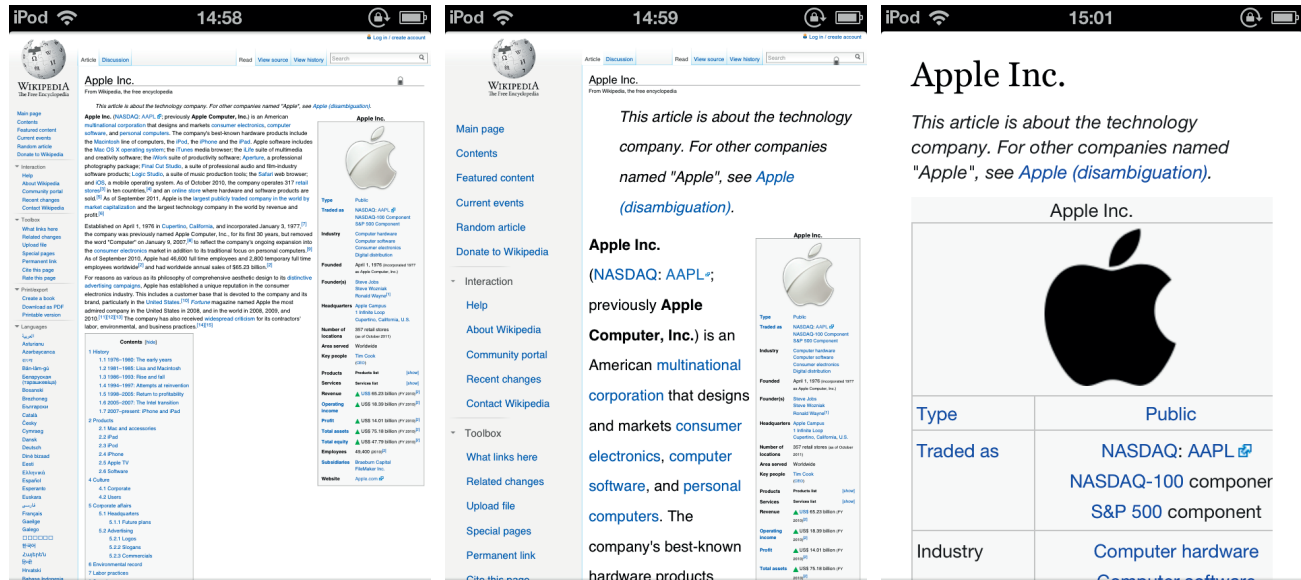


Figure 4: Different Wikipedia interfaces compared in the evaluation: (1) Desktop, (2) W3Touch-based Version, (3) Mobile

grouped and ranked according to importance and frequency as part of the interview transcription.

In the second step, we performed an experiment based on W3Touch to see whether the tool would help detect similar issues and perhaps reveal additional problems. To make for a controlled experiment, it was divided into two phases. In the first phase, we collected tracking data using W3Touch for an example article from the original Wikipedia. For the second phase, we asked two students in our group with web design experience to define new design rules based on the user performance data collected by W3Touch. We then tested the effects in another run of the experiment. The experiment had a total of 84 participants using a variety of touch devices. This part of the evaluation showed that the students were able to identify the same problems. The visualisation techniques implemented in W3Touch helped them to quickly gain an overview of the data and explore the problems specifically for both smartphones and tablets and for different orientations of each type of device. The experiment also produced a new version of the Wikipedia site using W3Touch with simple adaptations based on the average zoom and missed links metrics for further analysis.

In the third and final step, we compared the new W3Touch-based version against both the Wikipedia Desktop and Mobile versions in a follow-up lab study with 13 participants. Since the version based on W3Touch required relatively little design effort, we were especially interested to see how it fares against the manually designed mobile site. This part of the evaluation showed that our W3Touch version of Wikipedia significantly improves task performance and the perceived user experience levels compared to the desktop version. It also came very close to the results of the mobile site which is very promising given that W3Touch poses relatively little effort on designers. We provide more details below.

**Expert Interviews**

The original design of the Wikipedia article was reviewed by six experts in terms of the touch interaction and necessary adaptations. The interviewees were selected according to their background in web engineering and user experience design. Four participants were PhD candidates with active research and teaching roles in these fields, one had an industry background with several years of experience in web design and a computer science bachelor student who had recently completed an internship where they developed a number of touch and multi-touch prototypes for different mobile and desktop scenarios. Interviews took between 25 and 50 minutes. For the expert review, an iPod touch 2G and an iPad 1G (both running iOS 4) were used as popular examples of smartphones and tablets.

**Structure** The participants were presented the original Wikipedia Desktop on one of the devices (the device used first was alternated between interviews to counterbalance ordering effects). The interview started with a first general task where the participants were asked to thoroughly go through the example site and—while thinking aloud—identify potential issues in terms of the design, layout and touch interaction and state how they would adapt the design and layout to address potential problems. While this task was meant to find out where experts would see the primary issues, in a second task, they were asked to have a specific look at the different components of the page (e.g. header or footer) that were often not reviewed in the first step and to consider any differences between portrait and landscape mode. After the first round, the participants were presented the second device with the same tasks and asked to mention explicit differences in terms of the required adaptations. **Findings** In general, the experts stated that, independent of either device, the huge amount of scrolling caused by the length of the article text makes it difficult to get a good overview and that this would

be one of the primary issues. Moreover, they pointed out that the main component of interest would therefore be the article text, while the header, navigation and especially the footer (one interviewee stated that they “never click on that”) would be less important in terms of possible adaptations.

Concerning the iPod touch, all experts remarked that the initial view of the page would be too small and not useful without zooming. While they felt that the thumbnail of the page might first give a good overview, this might be less helpful for sites like Wikipedia with a single type of main content and a continuous flow of text. In addition, all but two experts specifically mentioned the amount of zooming required as a usability issue, as this makes it even more difficult to keep an overview and also leads to horizontal scrolling. Regarding the required adaptations, all experts felt that the article container needs adjustments in terms of the font size and also the spacing around densely packed links to “make sure that all the interactive things you can touch have a reasonable size.” All but one expert also noted that this is especially true for the sidebar navigation and footer where the finger still covers several links even when zoomed in to a readable size. The header of the page was generally considered to be mostly fine because the small number of links in this component have larger touch areas by design and also sufficient spacing due to the horizontal layout. Finally, two experts explicitly pointed out that there is not much of a difference in terms of the required adaptations between portrait and landscape mode on such a small device. Other more advanced ideas for the adaptation would involve changing the general layout to one single column that fits the device with everything vertically aligned similar to the mobile version of Wikipedia. This could then also involve rearranging the vertical navigation bar into a horizontal one at the top of the page. Other common suggestions were the dynamic adjustment of the text width to the current viewport to avoid horizontal scrolling as this is automatically done by some browsers on certain devices. While paging was often mentioned as a potential improvement, one expert also said that vertical scrolling with the thumb would be a convenient way for users to interact with long pages while reading on small-screen devices.

The general statement concerning the iPad was that “there’s more room for improvement” simply because of the larger size and screen real estate. The majority of experts remarked that the initial view of the site is almost optimal, as it involves way less zooming than on the smaller test device, but that the greater differences between portrait and landscape mode could afford different adaptations. For example, for portrait mode, two experts stated that the navigation could be moved to the top of the page to give the article container more horizontal space, while it should stay in place in landscape mode to avoid longer lines of text. As compared to the iPod touch, the font used in the different components was considered to be relatively small in portrait mode, but better in landscape mode. Similar to the small-screen device, more spacing around densely packed links in all parts of the page was suggested. Accordingly, the header bar was again considered to be the “best” part of the web site in terms of the touch interaction. Other suggestions included removing less

important items from the navigation, or to let the sidebar navigation float to reduce scrolling efforts. One expert suggested allowing users to adjust the font size dynamically using sliders. Again, dynamic content pagination was suggested as a means to ease scrolling.

In summary, the experts found that, for better support on touch devices, the web site would primarily require adjustments in terms of the font size and spacing around links and that these adaptations would be more important on smaller devices to counter excessive zooming. Moreover, they found that, although many of the proposed adaptations would apply equally for all components with the exception of the header, the focus should be on the article container as the main component of interest.

### **Remote Usability Study**

The user studies as part of our experiment based on W3Touch had two primary goals. The first was to collect interaction data using W3Touch for the original test page in a real usage scenario, checking for the primary issues detected by our tool support and how they would compare to the expert findings. The second goal was to define a suitable adaptation catalogue for W3Touch based on the collected data and test whether the adaptations would improve the user experience.

**Method** To gather tracking data for typical scenarios in a short amount of time, the experiment was specifically designed to involve all web site components (top navigation, sidebar navigation, article text, footer) and used three types of tasks (Figure 5). Click tasks scrolled to links on the page and marked them for activation. In read tasks, text paragraphs were scrolled into view and highlighted for reading. Time was counted until participants indicated that they had finished reading. For the find task, users were asked to find a link marked in the article text, starting from the top of the page. This was a control task to measure the effect of adaptations on scrolling effort when comparing the two experiment phases. Participants had to complete a set of 33 pre-defined tasks followed by a questionnaire. The zoom level was measured by the system, but only controlled by the users themselves.

**Results** Overall, 84 participants with touch devices were recruited via student mailing lists as well as social networking and bookmarking services. W3Touch detected 64 smartphones and 20 tablets of different models. To make data analysis practical, the tracking data was grouped by these two device types and further distinguished by orientation, making for four different contexts, while the context engine also distinguished different display sizes. 50 participants (42 male, median age 25) provided questionnaire feedback and indicated that they use their mobile devices several times a day for web browsing. 39 participants took part in Phase 1 where W3Touch was used to collect data for the two metrics before the adaptation. 45 then tested the adapted interface in Phase 2.

Figures 6a and 6b illustrate the findings for both phases concerning the article text component (other components are omitted due to similar results).

Phase 1 showed the highest zoom levels for smartphone users, on average lower values for tablets and minor differences be-



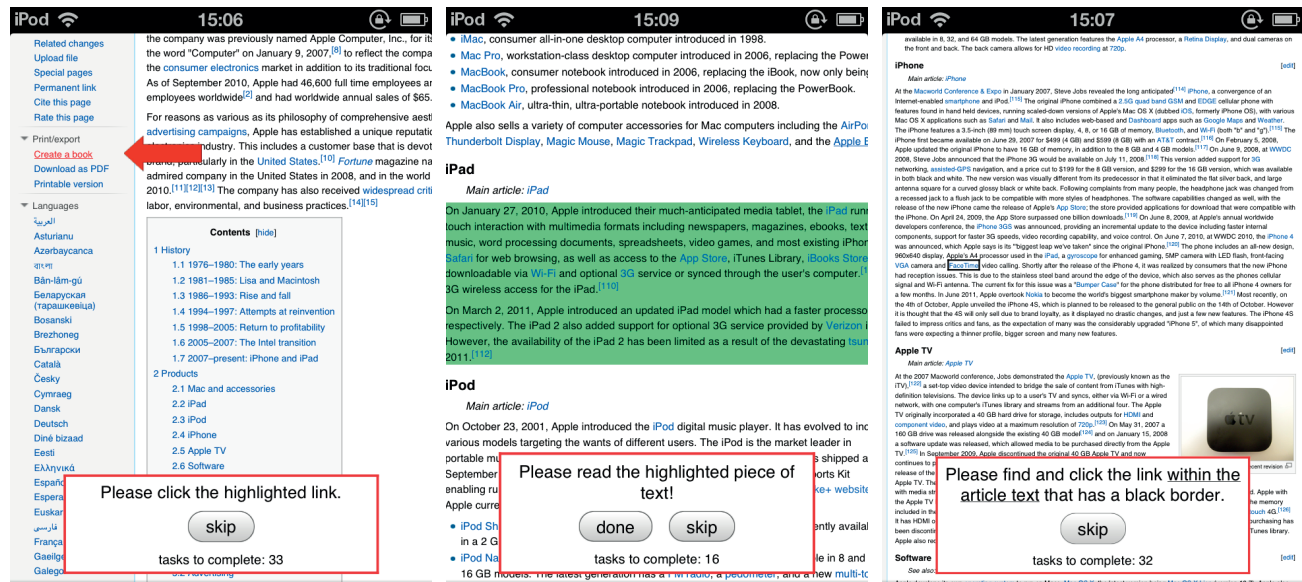


Figure 5: The set of tasks used in the experiment: users were asked to click pre-defined links and read selected text passages, while search tasks were used to check scrolling effort before and after adaptation (screenshots from Phase 1).

tween portrait and landscape mode. The missed links ratios were fairly high for both smartphones and tablets, with minor differences between portrait and landscape mode. The results produced by W3Touch therefore not only confirm the expert findings, but also deliver concrete numbers for different types of devices when it comes to potential interaction problems and the need for zooming. An interesting result that the experts did not consider is that there are also significant problems on tablets where the interaction seemed generally better.

For Phase 2, the students first explored the tracking data using W3Touch's visualisation tools for the four different settings. Similar to the expert recommendations, they decided to focus the adaptations on the sidebar navigation and article text using two relatively simple design rules. First, the font sizes were scaled according to the average zoom levels on a per-component basis, but to give most emphasis to the article text, the other components were weighted at only 67%. Second, the line height of links in the sidebar navigation and the article was increased by 33% for additional spacing relative to the increase in font size. Since the missed links ratios were generally high, the students did not specify thresholds in terms of this metric. Below we discuss some of the effects.

In terms of the average zoom for Phase 1 vs. Phase 2, Wilcoxon Signed Ranks tests confirmed significant differences on smartphones for all components (portrait:  $p < 0.001$ , landscape:  $p < 0.04$ ) as a positive effect of the increased font sizes. The differences on tablets were not significant. There were no significant differences between the two phases in terms of missed links, which could indicate that zoom level-based adaptations did not sufficiently increase touch areas to show an effect and that hyperlinks might need additional padding. However, users also zoomed less in Phase 2, e.g. 1.46 as opposed to 2.73 on smartphones, raising again

the chance of missing targets. The differences concerning the reading and answering times were not significant.

Despite the problems detected in Phase 1, the self-reported user experience was still fairly high. Phase 2 showed a positive trend for the reading feedback, but, as confirmed by Mann-Whitney tests, only the differences for the find link task on smartphones were significant ( $p < 0.01$ ) with a lower rating for Phase 2. This is what we expected due to the increased vertical scrolling effort and was considered an acceptable trade-off for reduced zooming and no more need for horizontal scrolling.

### Follow-up Lab Study

To see how close the version based on W3Touch comes to the mobile site specifically designed for smartphones, we conducted a lab study with 13 participants (9 male, median age 29) familiar with the Wikipedia site. We compared all three versions (Desktop, Mobile, W3Touch) on an iPod touch and used a similar set of tasks. However, we removed the find link task since Wikipedia Mobile integrates the table of contents with the text and collapses sections by default, which makes direct comparisons difficult. In terms of the average zoom levels (Figure 6c), k-samples Median tests showed significant differences for Desktop vs. W3Touch ( $p < 0.001$ ) and desktop vs. mobile ( $p < 0.001$ ). This generally speaks in favour of the adapted versions. However, the differences between W3Touch and Mobile were still significant ( $p < 0.02$ ). This could be explained by the fact that Wikipedia Mobile requires almost no zooming at all. There were no significant differences for the missed links ratios.

Our special interest was then in readability and how participants perceived the user experience (Figure 6d). As confirmed by pair-wise Wilcoxon Signed Ranks tests, both read-

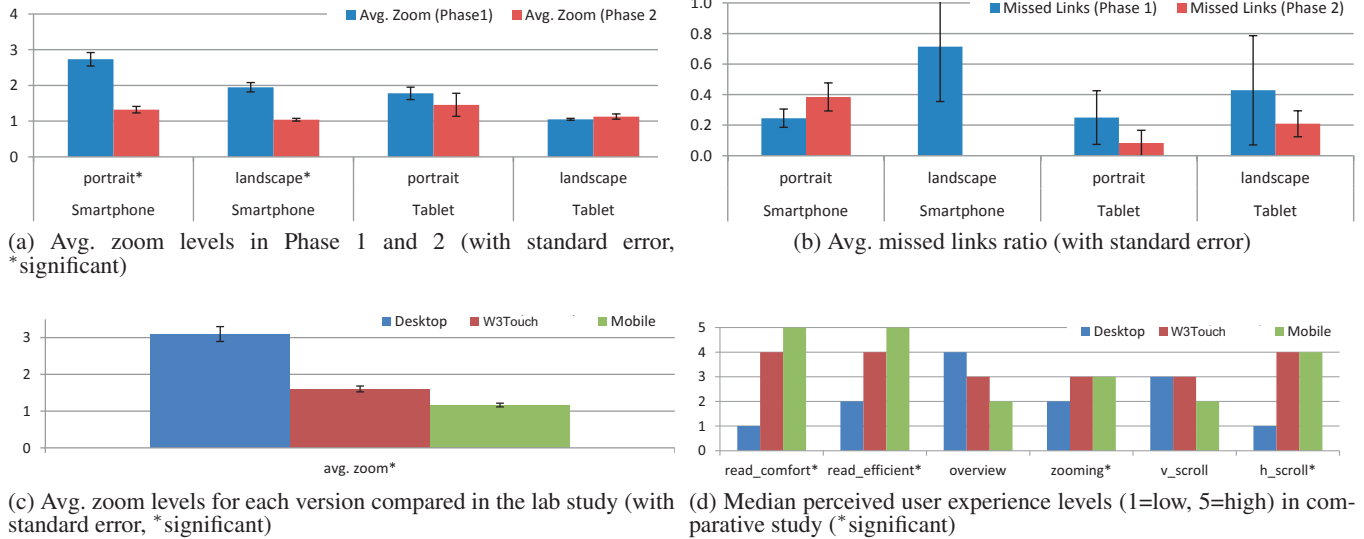


Figure 6: Results from the two user studies using W3Touch

ing comfort and efficiency were rated significantly higher than the desktop version for both W3Touch ( $p < 0.03$ ) and Mobile ( $p < 0.01$ ), with no statistical differences between the latter two ( $p > 0.15$ ). There were no significant differences in terms of reading and answering times. In terms of zooming, the adapted versions were generally better with significant differences for both Desktop vs. W3Touch ( $Z = -2.4$ ,  $p < 0.02$ ) and Desktop vs. Mobile ( $Z = -2.1$ ,  $p < 0.04$ ), but not for W3Touch vs. Mobile ( $Z = -0.4$ ,  $p > 0.66$ ). Concerning page overview and the required amount of vertical scrolling, there were no significant differences between the versions, which could suggest that neither version is optimal in this respect. Compared to the desktop version, horizontal scrolling was significantly better for both the W3Touch ( $Z = -2.7$ ,  $p < 0.01$ ) and Mobile versions ( $Z = -2.8$ ,  $p < 0.01$ ), with no significant differences for W3Touch vs. Mobile ( $Z = -0.8$ ,  $p > 0.39$ ).

Overall, the experiment showed that metrics-based adaptation using W3Touch is able to compete with a manually designed mobile site in several important aspects. This is very positive given the relatively small design effort. We therefore believe that the proposed tools are useful, but acknowledge that more detailed studies on web designers and formal user evaluations of W3Touch’s tracking and visualisation tools remain to be carried out in the future.

## DISCUSSION AND RELATED WORK

In the following, we will analyse the contributions of W3Touch in the context of related work. We will also discuss current limitations and ways of generalising W3Touch including the extension with new metrics in the future.

### Comparison to Existing Techniques

A popular way of characterising existing adaptation techniques is by distinguishing between adaptable interfaces and adaptive interfaces. Adaptable interfaces require the user to

perform the adaptation more or less directly, whereas adaptive interfaces are system-controlled relying on context information provided to the application [11]. Generally, existing approaches tend to focus on traditional WIMP rather than web interfaces and range from purely user-controlled techniques based on direct manipulation such as user interface façades [28] to fully automatic solutions like Supple [14] that can generate personalised interfaces using a rule-based approach or machine learning techniques. This paper investigated a new mixed-initiative approach specifically designed for web interfaces, introducing a new set of touch interaction tracking tools and metrics for controlling the adaptation.

The adaptation to different devices has often been dealt with in web engineering as one particular aspect of context-awareness. The focus of research in this discipline has tended to be on a systematic separation of concerns between content, navigation and presentation based on web design methodologies such as WebML [8] and Hera [13] in an effort to uniformly address the wide range of requirements of personalisation, location-awareness and multi-channel access. Most techniques are model-based to define interfaces at different abstraction levels and perform the adaptation to different user, platform and environment contexts based on a series of model transformation steps [7]. To this end, design strategies for adaptive and multi-modal interfaces such as graceful degradation [12] as well as user interface description languages, e.g. MARIA [27], have been the subject of extensive research. However, with the exception of multi-device design environments such as Damask [21] and Gummy [22], which offer a multi-layer editing approach and specific tools for designers to record, share and reuse adaptations for different devices, most approaches require significant technical knowledge and tend to come at the price of increased design complexity.

Other solutions allow for the interactive reauthoring of web sites, but are typically limited to very specific adaptation sce-



narios only. For example, PageTailor [4] provides a direct manipulation interface specifically designed for PDAs with simple controls for users to move, resize and hide elements in a web page. Highlight [26] allows users to demonstrate interactions with a web page that are then used to create a smaller version of the original page adapted for mobile applications. Another major limitation of the latter is that it does not support the customisation of elements in terms of the visual appearance and that it is designed to be performed in a desktop environment rather than directly on the mobile devices.

The latest work in this line of research is Bricolage [19], an algorithm for automatically retargeting web sites using existing designs as an example. Bricolage primarily builds on machine learning, but uses crowdsourcing to train the models used to identify semantically similar web page elements. To support the adaptation to touch, Bricolage could in principle learn a new cost model, but this would require a substantial amount of examples as input to train the model for many different kinds of touch devices.

The closest to W3Touch is perhaps Chickenfoot [5], a browser plugin that enables customisation of rendered web pages and automation for a variety of tasks. The focus of Chickenfoot is however on allowing end-users to script common web interactions within and across web sites based on the visual appearance of DOM elements using keyword pattern matching, as opposed to our focus on web designers and support for adapting interfaces to different touch devices.

W3Touch distinguishes itself by the fact that design rules are defined and triggered by the evidence that a particular use context causes usability problems. This is opposed to current practice of adapting for specific hardware settings directly. The main benefit of our approach is that the underlying techniques directly and continuously involve end-users so that designers get informed about individual user preferences and new requirements as soon as they emerge. For example, W3Touch's tracking tools could alert developers and make them aware of the need to cater for a new touch device that just appeared on the market. Another advantage of W3Touch is that, given a set of relative adaptation rules as in our experiment, web interfaces are basically able to self-repair by counterbalancing interaction issues based on real usage data. However, this may also require site-specific prioritisation in terms of the components to be adapted.

### Current Limitations and Possible Extensions

The paper has focused on two metrics for detecting missed links and zoom levels, but W3Touch could be configured to record additional information such as tracking the frequency of zoom operations. This could be done by registering a new metric `num_zoom` in W3Touch and implementing the relevant tracking functions in the form of callback handlers on the client and/or server side as required. Similar to the in-built `avg_zoom` metric, the new metric would get exposed via the `W3Touch` object in JavaScript so that developers could readily make use of the collected data at run-time.

While the basic tracking mechanisms of W3Touch provide a first automatic tool for detecting potential interaction issues,

our implementation currently does not consider the semantics of the content elements involved in the interaction. This could help to derive meaning from the interaction and structure the detected problems by associating a certain cost with each of them. For example, we could track a different kind of mis-click in cases where a user accidentally activates the wrong link and quickly tries to navigate back to the previous location, and give a higher penalty to such mis-clicks. The additional information could also be used for prioritisation when highlighting critical components and applying the fixes in terms of adaptation rules.

At the moment, W3Touch is mostly useful for detecting spatial issues of the layout in terms of the size and position of text, media and links in specific use contexts. While the tracking data collected by W3Touch already allows for a number of useful adaptations, the range of possible adaptations that can be more or less inferred from the interaction and automated are still constrained by what can be tracked with the help of input sensing techniques supported by the device in use. However, in combination with additional input technologies and advanced techniques for eye, finger and skeletal tracking, which are becoming more affordable and increasingly mobile, it should soon be possible to implement new mechanisms. This would allow W3Touch to be extended with additional metrics and aggregate functions for different purposes and largely automated problem handling strategies by similarly extending the adaptation catalogue.

Finally, we need to note that optimising only for some metrics such as the missed links and zoom levels might negatively impact other usability factors. As mentioned above, additional metrics and adaptation rules could be used to balance the different factors and mitigate such issues. Since W3Touch is designed to leave the control with web designers rather than relying on fully automatic approaches, they may use the different tools to visualise and act on other usability problems as required. For example, in our previous work [24], we have developed a set of metrics that can be used to assess the quality of web layout under different viewing conditions. The metrics presented there could be combined with the touch-specific ones developed in this paper to better manage with the limited screen real estate on mobile devices, while still making sure that all touch-sensitive interface components are appropriately sized for finger input.

### CONCLUSION

This paper introduced W3Touch consisting of a set of tools for automated usability evaluation and adaptation of existing web interfaces to cater for the specifics of touch interaction. W3Touch builds on existing usability evaluation methods and extends them so that user performance data can be collected and used to inform the design of new adaptations. By the example of an existing web site, we demonstrated that it is feasible to automate the adaptation process for different touch devices based on user interaction tracking data and simple adaptation mechanisms embedded in the interface. In the future, we plan more detailed evaluations of the benefits for both expert web developers and novice web designers. We are also currently working on extending the tool support with

additional tracking mechanisms based on a combination of multi-touch and 3D-gesture tracking techniques to study the applications for multi-device environments with both small and large interactive surfaces.

### Acknowledgements

We would like to thank our study participants. We also thank Rob Miller for his helpful comments on an earlier version of this paper. This work was supported by the Swiss National Science Foundation under research grant 200020\_134983.

### REFERENCES

1. Atterer, R., Wnuk, M., and Schmidt, A. Knowing the User's Every Move – User Activity Tracking for Website Usability Evaluation and Implicit Interaction. In *Proc. WWW* (2006).
2. Baudisch, P., Xie, X., Wang, C., and Ma, W.-Y. Collapse-to-Zoom: Viewing Web Pages on Small Screen Devices by Interactively Removing Irrelevant Content. In *Proc. UIST* (2004).
3. Benko, H., Wilson, A. D., and Baudisch, P. Precise Selection Techniques for Multi-Touch Screens. In *Proc. CHI* (2006).
4. Bila, N., Ronda, T., Mohamed, I., Truong, K. N., and de Lara, E. PageTailor: Reusable End-User Customization for the Mobile Web. In *Proc. MobiSys* (2007).
5. Bolin, M., Webber, M., Rha, P., Wilson, T., and Miller, R. C. Automation and Customization of Rendered Web Pages. In *Proc. UIST* (2005).
6. Buyukkokten, O., Garcia-Molina, H., and Paepcke, A. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In *Proc. WWW* (2001).
7. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonck, J. A Unifying Reference Framework for Multi-Target User Interfaces. *IWC* 15, 3 (2003).
8. Ceri, S., Daniel, F., Matera, M., and Facca, F. M. Model-driven Development of Context-Aware Web Applications. *TOIT* 7, 1 (2007).
9. Charland, A., and LeRoux, B. Mobile Application Development: Web vs. Native. *CACM* 54, 5 (2011).
10. Chen, Y., Ma, W., and Zhang, H. Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. In *Proc. WWW* (2003).
11. Findlater, L., and McGrenere, J. A Comparison of Static, Adaptive, and Adaptable Menus. In *Proc. CHI* (2004).
12. Florins, M., and Vanderdonck, J. Graceful Degradation of User Interfaces as a Design Method for Multiplatform Systems. In *Proc. IUI* (2004).
13. Fräsincar, F., Houben, G.-J., and Barna, P. Hypermedia presentation generation in Hera. *IS* 35, 1 (2010).
14. Gajos, K. Z., Wobbrock, J. O., and Weld, D. S. Improving the Performance of Motor-Impaired Users with Automatically-Generated, Ability-Based Interfaces. In *Proc. CHI* (2008).
15. Hattori, G., Hoashi, K., Matsumoto, K., and Sugaya, F. Robust Web Page Segmentation for Mobile Terminal Using Content-Distances and Page Layout Information. In *Proc. WWW* (2007).
16. Holz, C., and Baudisch, P. Understanding Touch. In *Proc. CHI* (2011).
17. Hong, J. I., Heer, J., Waterson, S., and Landay, J. A. WebQuilt: A Proxy-based Approach to Remote Web Usability Testing. *TOIS* 19, 3 (2001).
18. Karlson, A. K., and Bederson, B. B. One-handed touchscreen input for legacy applications. In *Proc. CHI* (2008).
19. Kumar, R., Talton, J. O., Ahmad, S., and Klemmer, S. R. Bricolage: Example-Based Retargeting for Web Design. In *Proc. CHI* (2011).
20. Lam, H., and Baudisch, P. Summary Thumbnails: Readable Overviews for Small Screen Web Browsers. In *Proc. CHI* (2005).
21. Lin, J., and Landay, J. A. Employing Patterns and Layers for Early-Stage Design and Prototyping of Cross-Device User Interfaces. In *Proc. CHI* (2008).
22. Meskens, J., Vermeulen, J., Luyten, K., and Coninx, K. Gummy for Multi-Platform User Interface Designs: Shape me, Multiply me, Fix me, Use me. In *Proc. AVI* (2008).
23. Nebeling, M. *Lightweight Informed Adaptation: Methods and Tools for Responsive Design and Development of Very Flexible, Highly Adaptive Web Interfaces*. PhD thesis, ETH Zurich, DOI: 10.3929/ethz-a-007562962, 2012.
24. Nebeling, M., Matulic, F., and Norrie, M. C. Metrics for the Evaluation of News Site Content Layout in Large-Screen Contexts. In *Proc. CHI* (2011).
25. Nebeling, M., and Norrie, M. C. Lightweight Toolkit and Development Framework for Multi-touch/Multi-device Web Interfaces. In *Proc. EICS* (2012).
26. Nichols, J., Hua, Z., and Barton, J. Highlight: A System for Creating and Deploying Mobile Web Applications. In *Proc. UIST* (2008).
27. Paternò, F., Santoro, C., and Spano, L. MARIA: A Universal, Declarative, Multiple Abstraction-Level Language for Service-Oriented Applications in Ubiquitous Environments. *TOCHI* 16, 4 (2009).
28. Stürzlinger, W., Chapuis, O., Phillips, D., and Roussel, N. User Interface Façades: Towards Fully Adaptable User Interfaces. In *Proc. UIST* (2006).