The Proxemic Web: Designing for Proxemic Interactions with Responsive Web Design

Ryan Sukale
School of Informatics & Computing,
Indiana University,
535 W. Michigan Street, IT 475,
Indianapolis, IN 46202 USA
rysukale@umail.iu.edu

Stephen Voida
School of Informatics & Computing,
Indiana University,
535 W. Michigan Street, IT 475,
Indianapolis, IN 46202 USA
svoida@iupui.edu

Olesia Koval
School of Informatics & Computing,
Indiana University,
535 W. Michigan Street, IT 475,
Indianapolis, IN 46202 USA
olesiakoval@iupui.edu

UbiComp'14 Adjunct, September 13-17, 2014, Seattle, WA, USA
ACM 978-1-4503-3047-3/14/09.
http://dx.doi.org/10.1145/2638728.2638768

Abstract
Responsive web design has become one of the guiding principles for delivering a consistent viewing experience for the same World Wide Web content across devices of different sizes. However, this principle covers only half of the spectrum of a viewing experience, since it does not factor in proxemic interactions between a user and a single device. In this research, we propose an approach for adapting the construct of responsive web design to account for proxemic interactions and provide an example implementation.

Author Keywords
Proxemics; design; web design; human factors; ubiquitous computing; responsive web design

ACM Classification Keywords:
H.5.2 [Information interfaces and presentation (e.g., HCI)]: User interfaces—interaction styles.

Introduction
Proxemics is the science of understanding the effect of a person's distance from their surrounding entities—people and objects—on their relationships with those entities. Understanding proxemic interactions is crucial

to the field of ubiquitous computing since distance tracking can allow for new modes of interaction with existing interfaces. The recent past has seen an increase in the number of end-user distance tracking devices, especially in the domain of gaming, exemplified by the Microsoft Kinect[1] and Nintendo Wii[2]. The market has also seen an increase in smaller and less expensive distance-based interaction devices like the LeapMotion[3], which enables gesture-based touchless interaction on ordinary desktop computers. As these devices continue to be integrated into laptop and desktop PCs, a significant opportunity will emerge to harness information about the user's proximity to enhance the web-browsing experience.

Responsive web design has now been accepted as the *de facto* strategy for content delivery in the web browser across devices with different screen sizes. In this poster, we propose the idea of considering proxemic interactions in the design of the next generation of responsive websites and demonstrate a prototype of how such a system might work.

Related Work

Truly responsive technologies demand a deep knowledge of the relationship between people and devices as well as the ways this relationship can affect interaction [1]. Ballendat and his colleagues explored the notion of leveraging both implicit interactions (i.e., action among end users) and explicit interactions (i.e., actions made by end user specifically to communicate with the system) to realize a proximity-aware system. They created a video media player that implicitly changes the displayed information depending upon people's proximity and orientation to the display. Marquardt and colleagues created an announcement board that presents information differently depending on the distance of a user from the board [5]. Prante and colleagues [6] created a wall-sized ambient display called *Hello.Wall* that worked in conjunction with mobile devices called Viewports to allow for 2-way communications between a user's devices and the wall. Although these efforts mainly focused on the interaction modalities of wall-sized displays, they strongly indicate how user proximity information could be used to enhance the capabilities of and modes of interaction supported by everyday software applications. An increase in the deployment of location-sensing technologies [3] make it possible to determine a user's distance from his/her device and present information appropriately based upon this data.

Researchers have previously explored the need for content magnification in desktop computing devices. Harrison and Dey created a system called Lean and Zoom [2] that takes into account people's natural behavior of leaning towards a screen when they need to inspect unseen details and automatically magnifies screen content in proportion to the extent of the lean.

Web browsers have been an integral part of the internet revolution. In the recent past, two important things have happened to dramatically change the browser landscape. The first was the explosion in the number of mobile devices. With more consumer-facing content now being delivered for mobile devices, businesses large and small have not only had to create new applications for their mobile consumers but also rethink

[1] http://www.xbox.com/en-US/kinect
[2] https://www.nintendo.com/wii
[3] https://www.leapmotion.com/

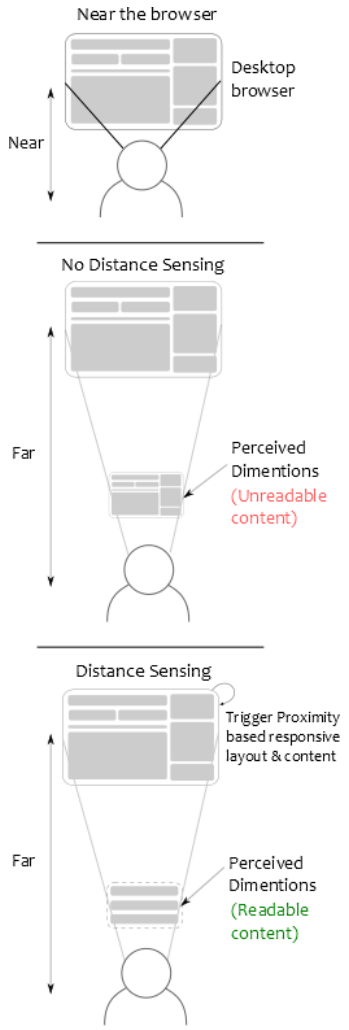


Figure 1. An overview of the benefits of the *proxemic web* approach: (top, middle) Existing information display experiences at different distances; (bottom) Our proposed solution.

how their existing websites would be presented on mobile devices' much-smaller displays.

To solve this problem, a 'responsive design' strategy was first proposed by Marcotte [4]. The term was derived from an emerging discipline called 'responsive architecture' whose primary concern was to design buildings that could respond to the presence of the people passing through them. Responsive web design approaches the web content adaptation problem by following a 'flexible grid' layout, thereby giving designers the option to create different layouts in different grid arrangements such that these different grid arrangements are used for different screen sizes. This technique eliminates the need to make design assumptions about the width of the browser window.

The second major factor currently re-shaping the browser landscape is an increase in browser–hardware integration [8]. A direct application of this kind of technology can be seen in the increasing sophistication of in-browser games. Hardware integration will allow web designers to tap into proximity-detection hardware and take advantage of this data to provide a more comfortable and adaptive web browsing experience.

The *Proxemic Web*

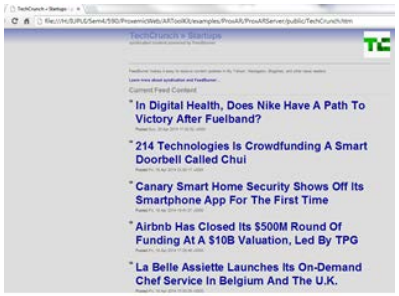
Considering the widespread adoption of responsive web design and an increase in location- and distance-sensing hardware, a new challenge appears on the horizon for the browser platform. Unlike current systems, distance-sensing devices like the Microsoft Kinect or Leap Motion allow the user to interact with the personal computer from a distance without having the touch the keyboard and mouse. We envision a future where such gesture-based devices would not only be used for specific applications but would also be used for performing ordinary tasks like browsing the web. Using such proximity detection technology alongside an ordinary desktop computer not only requires proxemic interactions to be considered for the desktop environment but also requires us to revisit the paradigm of responsive web design from a different perspective.

Prototype Implementation

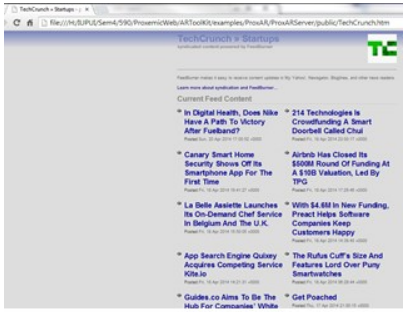
We created a system to adapt a web page's content based on the distance between the user and his or her desktop (Figure 1). For this prototype, to monitor a user's distance from the desktop, we used a standard webcam and ARToolKit[5]. The distance information was written to a file on the user's computer and fetched by the browser using Ajax. Although our prototype requires the user to wear a fiducial marker since we were using ARToolkit, inexpensive, commercially-available devices like the Microsoft Kinect or LeapMotion for example, can provide marker-less distance information.

We created a standard web page that incorporated a custom JavaScript library that periodically polls the distance information file to look for changes in the user–display distance. The CSS style sheets used to render the webpage are based upon three different distance thresholds—*near*, *medium* and *far* (Figure 2). In *near* mode, all content is displayed, along with pictures and text. As the user moves away from the screen and the script determines that a distance threshold has been crossed, it changes the layout to *medium*, removing any large images and rearranging the grid to display content at a larger size. Finally, when user moves even further from the screen, the layout changes to *far,* and only the

[5] http://www.hitl.washington.edu/artoolkit/



Screen content at *'far'* layout.



Screen content at *'medium'* layout.

Screen content at *'near'* layout.

Figure 2. Result of applying different CSS style sheets to a single web page, based on sensed differences in the distance between the user and the browser's screen.

title text is shown using large, single-line formats. We envision that our implementation could be generalized to support proxemic interactions across all websites by creating a browser plugin that detects proximity changes (or incorporating this capability directly into the browser) and connecting it to a CSS framework that controls webpage presentation based on distance.

Discussion and Future Work

Although we can borrow some concepts from the responsive web design strategy and apply them to a proxemic web design strategy, there are several challenges that will need to be overcome. First, we need to empirically determine how many distance thresholds are appropriate and what they are. Unlike smartphone vendors, there is no organization that controls the 'effective' dimensions of a device. Designers and developers across organizations would need to agree upon standardizing these thresholds in order for the consumer to have a consistent experience of proxemic interactions across different websites and devices

There is also the challenge of interaction design. Are the same interactions supported by wall-sized displays appropriate for browsing ordinary content at a distance? As far as interaction modalities are concerned, we expect that the same hardware used to determine the distance between the user and the screen might also be appropriated to detect gestural inputs, for example. For example, the Microsoft *Kinect* provides depth sensing by combining an infrared light source and an RGB camera. Together with the *OpenNI* platform, the Kinect can be used to recognize gestures—focus, swipe, wave and circle [7]. In our case, gestural swipes (up, down, left, or right) could be used to control *Proxemic Web* browsing at a distance. Further research is needed to assess the usability—and limitations—of this approach.

In a future where distance measurement devices are embedded into our desktop computers and browsers are able to take advantage of these devices, there will be a need for us to change the way we think about responsive web design if we are to use this information to create comfortable browsing experiences.

References

[1] Ballendat, T., Marquardt, N., & Greenberg, S. (2010). Proxemic interaction: designing for a proximity and orientation-aware environment. *Proc. ITS 2010*. 121–130.

[2] Harrison, C., & Dey, A. (2008). Lean and zoom: Proximity-aware user interface and content magnification. *Proc. CHI 2008*. 507–510.

[3] Hazas, M., Scott, J., & Krumm, J. (2004). Location-aware computing comes of age. *Computer, 37*(2): 95–97.

[4] Marcotte, E. (2010, May). Responsive Web Design. *A List Apart* [web magazine], issue 306. http://alistapart.com/article/responsive-web-design/

[5] Marquardt, N., Diaz-Marino, R., Boring, S., & Greenberg, S. (2011). The proximity toolkit: Prototyping proxemic interactions in ubiquitous computing ecologies. *Proc. UIST 2011.* 315–326.

[6] Prante, T., Röcker, C., Streitz, N., Stenzel, R., Magerkurth, C., van Alphen, D., & Plewe, D. (2003). Hello.Wall – Beyond Ambient Displays. *Adj. Proc. of Ubicomp 2003.*

[7] Villaroman, N., Rowe, D., & Swan, B., (2011). Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor. *Proc. 2011 Conf. Information Technology Education*, 227–232.

[8] Wiltzius, T., Kokkevis, V., et al. (2014, May). GPU Accelerated Compositing in Chrome. *The Chromium Projects*. http://www.chromium.org/developers/design-documents/gpu-accelerated-compositing-in-chrome