ZoneZoom: Map Navigation for Smartphones with Recursive View Segmentation

Daniel C. Robbins, Edward Cutrell, Raman Sarin, Eric Horvitz
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
{dcr, cutrell, ramans, horvitz}@microsoft.com

ABSTRACT

ZoneZoom is an input technique that lets users traverse large information spaces on smartphones. Our technique ZoneZoom, segments a given view of an information space into nine sub-segments, each of which is mapped to a key on the number keypad of the smartphone. This segmentation can be hand-crafted by the information space author or dynamically created at run-time. ZoneZoom supports “spring-loaded” view shifting which allows users to easily “glance” at nearby areas and then quickly return to their current view. Our ZoneZoom technique lets users gain an overview and compare information from different parts of a dataset. SmartPhlow is an optimized application for browsing a map of local-area road traffic conditions.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: Input Devices and Strategies, Interaction Styles

General Terms

Human Factors, Design

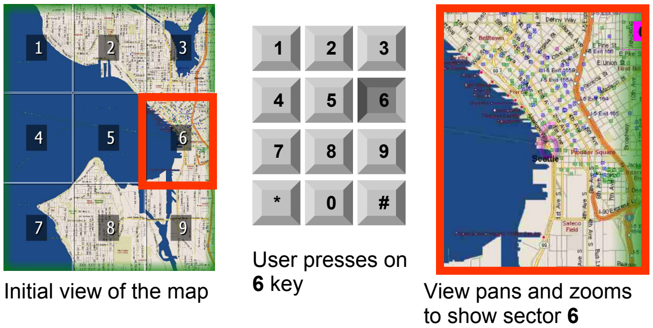
Keywords

ZoneZoom, SmartPhlow, visualization, hand-held devices, smartphones, spatial cognition, maps, mobile browsing, Zoomable User Interfaces, Visual Interaction

1. INTRODUCTION

Our technique ZoneZoom is geared toward the emerging class of mobile computing devices called smartphones. Smartphones combine the portability and network connectivity of cell-phones with the computing power of PDAs. Their limited input modalities make it difficult for smartphones to support both incremental and ballistic navigation (rapid navigation towards a target typically initiated by a command or button press) of large datasets. Because smartphones are often used in mobile situations where a user’s attention is divided, interactions that require precise control are challenging.

The size and layers of detail in common information spaces, such as maps, spreadsheets, and web pages, easily overwhelm the screens of small devices [[6]]. When a user is zoomed in far enough to see relevant detail, it becomes tedious to navigate across large distances using the d-pad of the smartphone. It is also difficult, when zoomed in, for the user to retain a sense of context and maintain a mental model of the information space. This paper details a combination of techniques, called ZoneZoom which adapt Zoomable user interfaces for map viewing on smartphones.



Initial view of the map

User presses on **6** key

View pans and zooms to show sector **6**

Figure 1: Zone Selection for Zooming

2. RELATED WORK

Previous small-device information space navigation research has focused on three main areas: ensuring that the content of a sub-section of the data-set can be easily read, providing simple navigation options for the user, and helping the user understand where a given sub-section is located relative to the entire information space.

When possible, content analysis is used to divide up the information space into areas of related content. When that is not possible, the content can be divided spatially such that any sub-section is guaranteed to fit on the display screen [[4], [6], [18], [5]]. Techniques that switch between a textual list of locales and views of a sub-section do not work well with maps, where users want to compare spatial information or make distance judgments.

Zoomable user-interfaces (ZUIs) arrange information in an infinitely pannable space and at many nested levels of detail [[3]]. Users navigate by use of a combination of panning and zooming. Various cues have been suggested for helping users find information in ZUIs [[11], [10], [9]] although effective visual design is even more difficult on the extremely small display of a smartphone. There is also a class of promising research that uses motion and tilt sensors to control navigation [[16]].

Fisheye techniques [[7]], which compress non-focused parts of the information space into the periphery, may not work well for maps. This is because users rely on geometric relationships to make judgments about travel time. Other work has explored adding schematic peripheral cues to indicate important off-screen information [[1], [12]].

3. ZONEZOOM

Our intent is to let users easily, quickly, and predictably use one hand to navigate around an infinitely detailed 2-dimensional information space on a smartphone. We have focused on the navigation of maps in general and the browsing of a near-real-time road traffic monitoring application in specific. As with previous work in this area we address the issues of information space segmentation, navigation, and context.

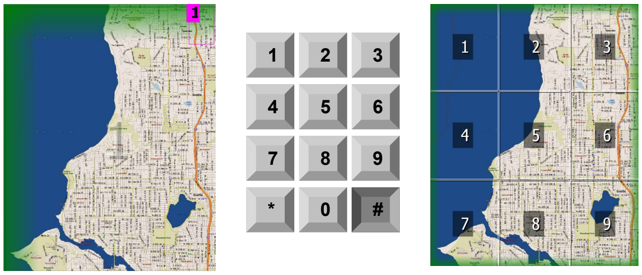
Because there is no direct pointing device on a smartphone (such as a stylus or mouse) there is no way for a user to directly specify an exact location on a map. The fine-grained interaction required by ZUIs in turn requires a high degree of attention on the part of the user – and attention is a scarce resource in mobile situations. With ZoneZoom we combine the interactions of panning and zooming into one "cognitive chunk" [[8]].

In the simplest implementation, a given view of a map at a particular zoom level can be evenly divided into a set number of sub-sectors or child views. For a smartphone, the natural number of sub-sectors is nine, since this corresponds to the number of buttons on the number pad. A set of outlined boxes with a numeric cue is superimposed over the map to indicate how the current view is segmented into child views. The user can then initiate an animated zoom into one of these sub-sectors by pressing on the number key that corresponds to that sector of the screen [Figure 1]. Pressing the same number key again, after the zoom-in action, will toggle the view to zoom back out to the parent view. If zoomed in, pressing a different number key will cause the view to gracefully pan to the appropriate sibling sector at the same zoom level.

ZoneZoom augments the precise movements of the D-pad with large ballistic movement that allows a user to quickly traverse large areas of the information space [[8]].

3.1. Arbitrary Levels of Zoom

To extend the view segmentation to arbitrary levels of zoom, we let the user display and make available the child sectors of the currently zoomed-in view by pressing a dedicated "toggle children" button on the smartphone keypad (in our implementation, the "#" key). When the user presses this key, several things happen: nine new view segmentation boxes are displayed; the current view is redefined to be the parent view for these newly available children views; and subsequent number key presses predictably zoom in to the appropriate child views. If the user is zoomed-in and wants to then zoom back out from the current view, the user can either press the number key that corresponds to the current view or press on a dedicated "zoom-out" key (in our implementation, the "*" key). Pressing the zoom-out key causes the view to zoom out and displays the child view boxes for the new current view [Figure 2].



View is zoomed-in to sector **1**

User presses on **#** key

Current view is segmented to show nine children views

Figure 2: Arbitrary Levels of Zoom

3.2. Visual Cues

Because ZoneZoom is an adaptation of a ZUI, we face the same difficulty in helping a user keep track of their location in the information space. Numbers are semi-transparently superimposed on each sector to assist users in associating a particular key with the appropriate sector. Because only a small number of sub-sector cues are shown at any time, the concept of preattentive discrimination [[17], [8]] suggests that all of the visible sub-sectors can be processed by the user at a single glance.

When zoomed in, the current sector number is made more transparent so that it does not obscure as much of the content. In addition, a very small and schematic overview map is superimposed in outline form in the upper right corner of the display. A smaller rectangle (the "detail view indicator") within the overview indicates the relative size and location of the current view within the context of the current view's parent view. This is akin to the transparent overview and ZUI combination seen in previous systems [[15]]. A small filled rectangle indicates the location of the currently viewed sector relative to its parent sector. A subtle, but noticeable gradation is applied around the entire margin of the parent sector. Thus, when zoomed in, a subsection of the gradated margin reinforces the location of the currently viewed child sector relative to its parent [Figure 1].

Our overview and its embedded detail view indicator only show the relationship between the current child view sector and its parent view. We surround each cue (such as a sub-sector number) with a contrasting border and vary its level of transparency depending on the current interaction.

3.3. Support for "Glance" Gestures

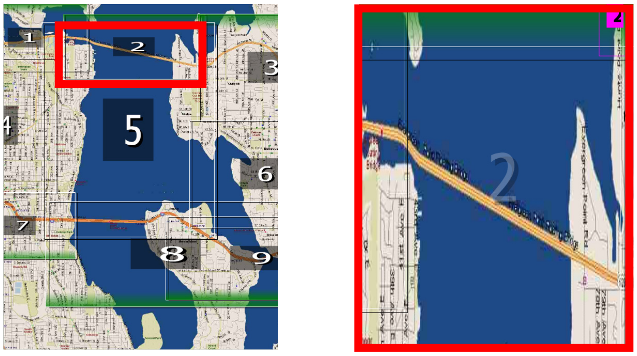
In ZoneZoom, we provide users with a quick means of building up contextual information about a current view. If instead of quickly tapping, the user holds a key down for more than a moment then releases (press-and-hold), the current view returns to the view from before the key press. If the user is currently zoomed out, such that segment cues are shown for the current view's child segments, pressing and holding on the number key will temporarily zoom the view to the appropriate child view. Releasing that same key will then return to the parent (zoomed out) view.

This aspect of spring-loaded view shifting is also seen in consumer flight simulators applications [[13]] where a spring-loaded hat-switch on a joystick lets the user momentarily glance at different views from the airplane's virtual cockpit.

This spring-loaded view shifting allows the user to quickly glance at other sections of the dataset without losing track of their preferred center of interest. This glancing also lets a user quickly compare information in nearby regions.

3.4. Non-regular Division of Zoom-levels

The aspect ratio of view sub-segments can differ from that of their parent or sibling views in order to devote more screen space to important features. When transitioning between view sub-segments that have different aspect ratios, the underlying map will appear to stretch and squeeze along its two cardinal axes [Figure 3]. In our current implementation, these optimized child view sectors are hand authored.



Sub-sectors are of various aspect ratios to optimize display of features in the map

During the view animation, the map shrinks and stretches so that the aspect ratio of the selected child view (**2**) fills the frame

Figure 3: Non-regular Division of Zoom-levels

4. IMPLEMENTATION

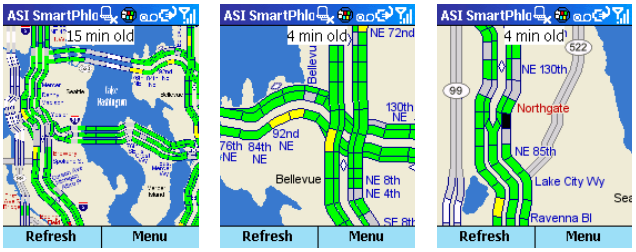
4.1. Prototype

ZoneZoom has been implemented on two different platforms. For prototyping we used Macromedia Flash MX on a desktop PC. The Flash prototype uses a high-resolution map of the Puget Sound area of Washington State that shows detail all the way down to the street level.

4.2. SmartPhlow Traffic Monitoring Application

Existing web based traffic map visualizations are ill suited to smartphone navigation [[14]]. Simultaneous with development of the Flash prototype, we integrated parts of the ZoneZoom technique into a road traffic monitoring application for a smartphone, called SmartPhlow. The SmartPhlow application was developed using the Microsoft Smartphone 2002 SDK. Our own server connects regularly to the Washington State Department of Transportation (WSDOT) traffic server [[20]]. When a user requests an update on their smartphone, the data is retrieved from our own server (via http over a GPRS or CDMA PCS network), thus reducing the load on the WSDOT server. The application downloads a relatively small file, usually on the order of 250 bytes, and expands it to populate a simplified map with color coded road segments [Figure 4].

In our current implementation, one line of text, overlaid at the top of the display, shows in relative terms the recency of the last update. Our user base typically has extensive experience with the web-based version of these maps so color-scheme interpretation is very quick. In the current version, the left soft-key is dedicated to refreshing the data and the right soft-key brings up a menu with additional options, such as auto-download and choice of map. The animation runs in a separate thread so that the user can choose to view another sector even before the current navigation animation finishes.

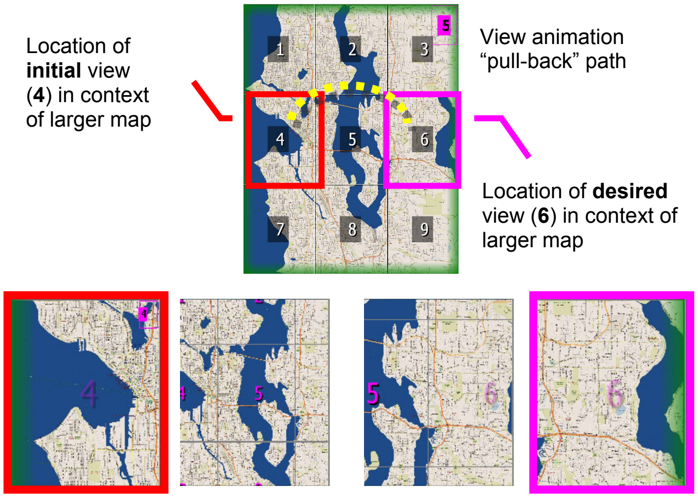


Zoomed-out view of the traffic map

Two different zoomed-in view sectors

Figure 4: SmartPhlow Traffic Monitoring Application

In SmartPhlow, only two levels of zoom are provided. As with the Flash version of ZoneZoom, pressing on different keys on the number pad moves the view among the nine sub-sectors. Both the Flash and SmartPhlow versions use a simplified version of the graceful transitions from van Wijk and Nuij [[19]]. All of the view shift animations are interpolated such that midway through the animation both the source (extents of the current view) and the destination (extents of the next view) can be simultaneously seen [Figure 5].



Location of **initial** view (**4**) in context of larger map

View animation "pull-back" path

Location of **desired** view (**6**) in context of larger map

View zooms out during pan from one sibling view (**4**) to another (**6**)

Figure 5: Animated View Transitions

5. DISCUSSION AND FUTURE WORK

SmartPhlow has been released internally within our organization. Feedback on an internal email list has been extremely positive, so much so, that the most common requests has been to add data for other locales and the ability to input personalized routes. In the future we will engage in formal user studies along with analysis of usage patterns (based on server statistics).

Maps exhibit many of the same properties as other information spaces: they are continuous, have a heterogeneous distribution of interesting data, possess multiple levels of detail, and usually have a highly variant visual representation [[10]]. Because of this, we intend to investigate the applicability of the ZoneZoom technique to other kinds of information spaces such as web pages, spreadsheets, and image browsing.

As noted in the Visual Cues section, the overview cues are of limited utility when a user navigates between more than two levels of zoom. We would like to design cues that scale well for indicating the level and position of the current view.

Finally, more work must be done to help users recover from accidentally hitting the wrong key. We need to show cues that help a user determine if the view has shifted in the desired way.

6. CONCLUSIONS

In this paper we have presented ZoneZoom, a technique for quick navigation of two-dimensional information spaces on smartphones. The information space is divided up into nested sub-segments, each of which is accessible via the number keypad. ZoneZoom has been adapted to a traffic map monitoring application called SmartPhlow and deployed to over 1000 users. Various visual cues, such as a schematic overview, numeric feedback, and gradated border from a parent view are used to reinforce the location of the current view. Future work for ZoneZoom and SmartPhlow includes the ability to customize the view segmentation and feature selection.

7. ACKNOWLEDGMENTS

We thank the Washington Department of Transportation (WSDOT) for providing the real-time traffic data and members of the internal SmartPhlow email alias for their useful feedback and suggestions.

8. REFERENCES

[1] Baudisch, P. and Rosenholtz, R. Halo: A Technique for Visualizing Off-Screen Locations. In *Proc. CHI'03*, pp. 481–488.

[2] Bederson, B.B., Czerwinski, M. & Robertson, G.. A fisheye calendar interface for PDAs: Providing overviews for small displays. In *The Craft of Information Visualization: Readings and Reflections*, pp. 324–344.

[3] Bederson, B., J. Hollan, K. Perlin, J. Meyer, D. Bacon & G. Furnas. Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, (7) 1996. pp. 3 – 31.

[4] Bjork, S., L. Holmquist, et al.. WEST: A Web Browser for Small Terminals. In *Proc. UIST'99*. pp. 187–196.

[5] Buyukkokten, O., Garcia-Molina, H. et al. Power Browser: Efficient Web Browsing for PDAs. In *Proc. CHI'00*. pp. 430–437.

[6] Chen, Y., W.-Y. Ma, et al. (2003). Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. In *Proc. WWW'03*, pp. 225 – 233.

[7] Furnas, G. W. Generalized Fisheye Views. In *Proc. CHI'86*, pp. 16–23.

[8] Hinckley, K., R. Jacob, C. Ware. Input/Output Devices and Interaction Techniques. In *CRC Computer Science and Engineering Handbook*, A. B. Tucker, ed. CRC Press LLC: Boca Raton, FL. to appear.

[9] Holmquist, L. E. and Ahlberg, C. Flip-zooming: A practical Focus+Context Approach to Visualizing Large Information Sets. In *Proc. HCI International '97*, pp. 763–766.

[10] Hornbæk, K., Bederson, B. B., & Plaisant, C. Navigation Patterns and Usability of Zoomable User Interfaces With and Without an Overview. *ACM Transactions on Computer-Human Interaction*, 9(4):362–389.

[11] Jul, S. & Furnas, G. W. Critical zones in desert fog: aids to multiscale navigation. In *Proc. UIST'98*, pp. 97–106.

[12] Mackinlay, J., Good, L., Zellweger, P., Stefik, M., and Baudisch, P. City Lights: Contextual Views in Minimal Space. In *CHI'03 Extended Abstracts,* pp 838–839.

[13] Microsoft Flight Simulator 2004, Microsoft Corporation http://www.microsoft.com/games/flightsimulator/.

[14] MSN Traffic reports, Microsoft Corporation, http://autos.msn.com/everyday/trafficreport.aspx

[15] Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E.. Context and interaction in zoomable user interfaces. In *Proc. AVI'00,* pp. 227–231.

[16] Rekimoto, J. Tilting Operations for Small Screen Interfaces. In *Proc. UIST'96*, pp. 167–-168.

[17] Treisman, A. Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*. Proc. 31/2, pp. 156 – 177.

[18] Trevor, J., Hilbert, D. M., Schilit, B. N., & Koh, T.K. From Desktop to Phonetop: A UI for Web Interaction on Very Small Devices. In *Proc. UIST '01*, pp. 121–130.

[19] van Wijk, J. J. and Nuij, Wim A. A. Smooth and efficient zooming and panning. In *Proc. INFOVIS'03*. pp. 15–22.

[20] Washington Department of Transportation Traffic Maps, http://www.wsdot.wa.gov/PugetSoundTraffic/