StripeMaps: Improving Map-based Pedestrian Navigation for Smartwatches.
Peer-reviewed author version

StripeMaps: Improving Map-based Pedestrian Navigation for Smartwatches

Dirk Wenig
Digital Media Lab
TZI, University of Bremen
dwenig@tzi.de

Johannes Schöning
Expertise Centre for Digital Media
Hasselt University – tUL – iMinds
johannes.schoening@uhasselt.be

Brent Hecht
Dept. of Comp. Sci. and Engineering
University of Minnesota
bhecht@cs.umn.edu

Rainer Malaka
Digital Media Lab
TZI, University of Bremen
malaka@tzi.de

ABSTRACT

Map applications for smartwatches present new challenges in cartography, a domain in which large display sizes have significant advantages. In this paper, we introduce *StripeMaps,* a system that adapts the mobile web design technique of linearization for displaying maps on the small screens of smartwatches. Just as web designers simplify multiple column desktop websites into a single column for easier navigation on mobile devices, StripeMaps transforms any two-dimensional route map into a one-dimensional "stripe". Through a user study, we show that this simplification allows StripeMaps to outperform both traditional mobile map interfaces and turn-by-turn directions for pedestrian navigation using smartwatches. In addition to introducing StripeMaps, this paper also has a secondary contribution. It contains the first empirical comparison of different approaches for pedestrian smartwatch navigation and illuminates their pros and cons.

Author Keywords

Smartwatches; Cartography; Mobile Maps; Pedestrian Navigation

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces—*input devices and strategies, interaction styles*

INTRODUCTION

Maps are expected to play an important role in the day-to-day use of smartwatches, with map apps included by default with the Apple Watch and with Google's Android Wear platform. Indeed, smartwatch maps have important benefits over their smartphone equivalents, particularly in the context of pedestrian navigation.

For instance, using map apps on a phone in this context requires keeping the phone in one's hands at all times, while both hands can remain mostly free when navigating via smartwatches. However, before smartwatch map apps can meet their potential, important cartographic challenges must be addressed.
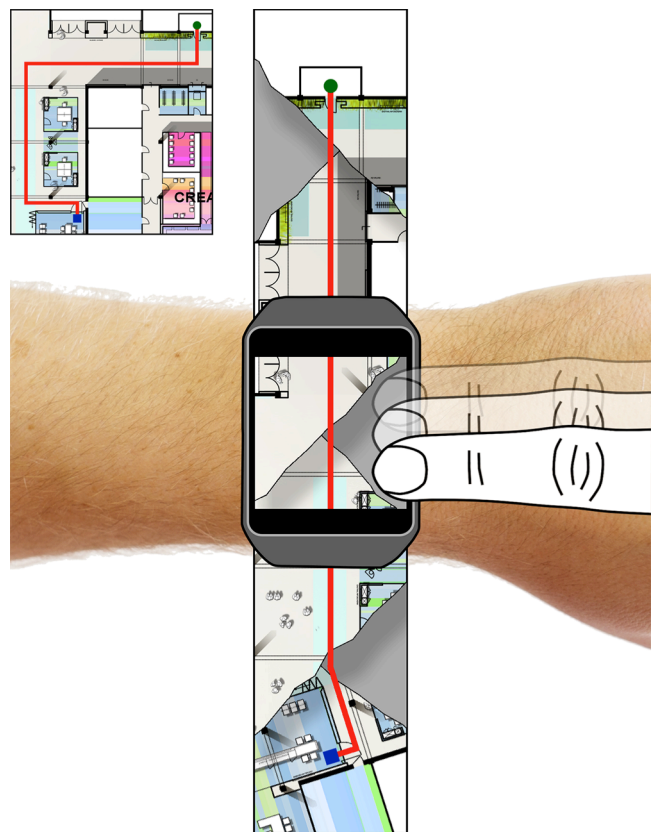

Figure 1: The StripeMaps concept. As screen space and interaction possibilities are limited on smartwatches, the StripeMaps application converts a 2D map to a 1D stripe. The original path on the 2D map is shown on the mini-map in the upper left corner. The cut (shown on the smartwatch) indicates the direction of the turn the user needs to make to navigate along the path.

These challenges arise from the very small screen size on smartwatches of typically 14 to 20 cm^2 (only a tenth to a quarter the size of a typical smartphone screen). Much of the art and science of reference map cartography (the type of cartography used in online and mobile maps [33]) involves simplifying a large, complex world for display on a much smaller canvas [19]. In general, the smaller the canvas, the harder the simplification – regardless of whether the canvas is digital or paper. Since smartwatch map apps are likely to be the smallest maps that have ever come into common use, existing simplification approaches may not work well and new cartographic techniques may need to be developed for important use cases.

In this paper, we introduce one such technique: *StripeMaps,* a novel cartographic approach for smartwatch maps targeted specifically at pedestrian navigation. StripeMaps is simultaneously motivated by the human-computer interaction (HCI) and cartography communities. Within HCI, the problem of adapting interfaces originally developed for larger-screen displays to smaller devices like mobile phones is well-known and well-studied [17, 12, 32, 36]. One best practice that has emerged in modifying desktop websites for mobile devices is *linearizing* the design from a multi-column layout to a layout with a single column[1]. Like desktop websites, maps extend in two dimensions, and the objective of StripeMaps is to adapt this linearization process from mobile web design to the cartography context.

The specific linearization approach taken by StripeMaps is motivated by a small family of traditional paper-based cartographic products used for long overland trips (e.g. via car or motorcycle). These products take a route which zigzags in two-dimensions and carve the route into segments such that each segment of the route can be rotated and displayed roughly horizontally or vertically for printing on a piece of paper. These sheets of paper, each with a more or less horizontal or vertical route segment (along with surrounding land area) are then arranged in order in a booklet. Probably the most well known application of this approach is the American Automobile Association's (AAA) TripTik map booklet. These route-customized booklets were commonly used on long road trips in the United States prior to GPS and smartphone navigation.

StripeMaps adapts this TripTik-style approach to linearize route maps into "stripes" (Figure 1). These "stripes" can be easily browsed on a smartwatch by scrolling in only one direction (as one does with a well-designed mobile website). By reducing a route from two-dimensions to one-dimension, StripeMaps simplifies the world into a single linear stripe, centered on the route of interest. In doing so, we hypothesized that StripeMaps would be better for pedestrian navigation on smartwatches than traditional

mobile map interfaces, which were originally designed for larger display devices and preserve both dimensions (and the corresponding additional complexity). At the same time, we also hypothesized that StripeMaps would have important pedestrian navigation advantages over text turn-by-turn directions, which eliminate all spatial context.

To test these hypotheses, we conducted the first user studies to compare cartographic approaches for smartwatch maps, in this case focusing on pedestrian navigation specifically. The results of these studies provide strong support for both of our hypotheses. When navigating using StripeMaps, participants completed complex indoor routes significantly faster than they did with (1) standard mobile map cartographic approaches on a smartwatch and (2) simple text turn-by-turn directions on a smartwatch. In addition, participants made significantly fewer errors in the StripeMaps condition than the turn-by-turn condition (and around the same number as in the traditional mobile maps condition). In an outdoor study that focused on more qualitative aspects, we also showed that StripeMaps can serve as an effective tourism companion. StripeMaps also scored significantly higher on a number of standard usability metrics than either baseline.

These results problematize some of the cartographic design choices of the first wave of smartwatches. Apple, for instance, currently uses traditional mobile map cartography in its Apple Watch map app[2]. This is an approach that our results suggest will lead to slower navigation times (for pedestrian navigation) and worse usability metrics than StripeMaps.

In summary, this paper makes the following contributions:

(1) We introduce StripeMaps, a new cartographic approach for smartwatch maps that results in improved pedestrian navigation relative to traditional mobile map cartography (the current de facto standard in smartwatches) and turn-by-turn directions.
(2) We perform the first empirical studies that compare smartwatch cartographic techniques. This sheds light not only on the advantages of StripeMaps, but also on the comparative pros and cons of existing cartographic approaches for smartwatch maps.

Below, we first introduce related work. We then detail the process by which StripeMaps converts 2D route maps into 1D stripes, and discuss how we implemented this process into a smartwatch application. We next cover the user studies, which demonstrate that StripeMaps is an improvement over other cartographic techniques in a number of important ways. Finally, we conclude with a discussion of open problems in this area and the ways we are seeking to address them in ongoing work.

---

[1] http://www.nngroup.com/articles/scaling-user-interfaces/

[2] http://gizmodo.com/apple-watch-will-give-you-a-buzz-when-its-time-to-turn-1632557384

RELATED WORK

In addition to the high-level motivations from mobile web design and cartography, this work also draws from and builds on research in the domains of (1) mobile navigation and wayfinding and (2) smartwatch interaction. We discuss related work from each of these domains in turn below.

Mobile Guides and Navigation

Work on map-based mobile guides dates back almost two decades (see Baus et al. for an early overview [4] and Huang et al. [11] for a more recent survey on mobile (indoor) navigation systems). Interestingly, some of the early map-based mobile guides made use of wrist-worn map interfaces. For instance, the DeepMap system [16], which was first published in 2000, presented animated route information on a wrist-mounted display.

Other early systems relied on pre-smartphone mobile devices to assist the user while navigating. The Cyberguide [1] showed a schematic map of one's surroundings, which was automatically updated based on the user's position. Similarly, the GUIDE [8] project involved developing a mobile tourist guide for the city of Lancaster, UK. The system allowed the user to choose between an overview map and a map of the local area on a mobile device.

Visualizing navigation routes and instructions is a critical aspect of the design of mobile navigation systems, and many approaches have been developed for doing so. Kray et al. [15] distinguish between textual and spoken instructions, 2D route sketches including simple arrows, 2D maps and pseudo realistic instructions such as 3D maps in different levels of abstraction. It was recommended that the choice of the instructions depend on the availability of location, orientation, cognitive resources and technical resources. Similar, Puikkonen et al. [26] evaluated different map designs for indoor navigation, whereas Schöning et al. [30] inspect a large dataset of publicly-displayed local maps to identify design decisions made by cartographers to inform the design of online and mobile maps. Recently, Alvina et al. [2] presented RouteLenses. They aim to make it easier for users of online maps to follow map itineraries by dynamically adapting properties of the motor space, based on both cursor position and route geometry.

Besides different visualizations of route descriptions and maps, developing novel interfaces to support indoor and outdoor navigation is an important research field within the area of HCI. For example, Mulloni et al. [22] presented a design of a mobile augmented reality interface to support indoor navigation. Recently, Möller et al. [21] evaluated different mobile user interfaces containing virtual and augmented reality elements. Nurminen et al. [23] investigated navigation via 3D maps on mobile devices, focusing on degrees of freedom. They concluded that the main challenge of designing 3D user interfaces is a tradeoff between free movement and efficiency achieved by limited and guided navigation. Wenig et al. [34] combined maps and images in a virtual 3D environment and used a pitch gesture to switch between the two media. To overcome small screen problems that occur when displaying maps on mobile devices, Reilly et al. [27] focused on how to use mobile devices with paper maps to express queries and present dynamic information for navigation and planning purposes.

Besides the use of mobile devices to support navigation, researchers and practitioners have also explored navigation on wearable devices, although none have investigated the efficacy of different cartographic approaches. Pielot et al. [25] explored the use of a vibrotactile belt to continuously indicate a destination's direction relative to the user's orientation. McGookin et al. [20] focused on supporting undirected navigation for runners in a system called *RunNav,* which could also be used on a smartwatch. Rather than offering explicit routes, *RunNav* provides a high-level overview of an area to allow for serendipity while at the same time informing runners of areas that are generally good and bad places to run. On the practitioner side of things, Android Wear is not the only wearable platform developed by Google that supports navigation: Google Glass also has built-in support for visualizing turn-by-turn directions.

Small Screen Visualization and Interaction

As screen space is very limited on smartwatches and the "fat finger problem" [32] heavily impacts the interaction, most work regarding smartwatch interactions explores additional input techniques or modalities [7, 14]. For example, Rekimoto [28] investigated the use of tiling as an additional input modality. As a smartwatch worn on the wrist of the user cannot be turned in all directions, his finding could fully be applied to smartwatches. Similar, the work of Xiao et al. [35] is concerned with expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. To extend the input space on smartwatches, Ashbrook et al. [3] investigated in depth the interaction on a circular touchscreen wristwatch. Oakley & Lee [24] address the "fat finger problem" by sensing touches to the perpendicular edges of a device. Kerber et al. [13] demonstrated how smartwatches could be used as magic lenses to browse through maps, but also showed that this technique did not outperform a "classical" UI.

THE STRIPEMAPS TECHNIQUE

The goal of the StripeMaps technique is to convert any 2D route map into a single 1D stripe (as in Figure 1). There are two major challenges involved in this process. The first involves *selecting a visualization strategy* for representing a 2D route in a straight line on a smartwatch. The second challenge involves *executing that transformation* on arbitrary 2D route maps.

Visualization Strategies

A number of dimension reduction strategies are available in the cartography literature and in cartography practice. The most well-known come from map projections, in which the goal is to minimize high-cost distortion when representing the earth's three-dimensional shape on a two-dimensional paper or digital surface.

The first visualization strategy we tested for StripeMaps was to transform the area along a route using an adaptation of the transmogrification approach by Brosz [6]. However, as can be seen in Figure 2a, these transformations are very difficult to interpret because it is hard to identify the decision points. In addition, labels and the environment at decision points can be heavily distorted. We also added a regular grid to the original map to overcome these effects, but with limited success. The complex transformation model was still hard to grasp for the users.

The underlying idea of the second approach, inspired by the work of Sarkar [29], was that the map could be "virtually printed" on a highly deformable material (such as fabric or clay) and then deformed into a stripe. An example visualized with the help of Blender can be seen in Figure 2b. This approach resulted in the less information loss along the way, but again the information density at decision points was very high.

Another alternative approach occasionally taken in cartography practice involves the "straightening" of mostly-linear features. For instance, this is done in the New York Times visualization of the Tigris and Euphrates rivers[3]. However, this approach only works well when the 2D feature being visualized has only soft curves and no sharp angles, something that is not the case for most pedestrian navigation routes. A similar approach is also employed in the TripTik maps discussed above. However, because each "slide" of the map appears on its own piece of paper, this approach does not address the important issue of how to integrate "slices" of the map into a single, continuous visualization.

Therefore, our third design adapted the core idea of the second approach, but instead of taking "fabric" or "clay", we used "paper". As analog maps are usually printed on paper, cutting a digital map into pieces and then rearranging the pieces into a stripe should be an easy to understand transformation model for the users. Furthermore, as can be seen in Figure 1 and Figure 2c, it results in less information loss at decision points. A side effect was also that the cuts provide excellent information on how to turn at the decision points. The cuts not only show the coarse direction but also the exact turning angle.

We compared these different *visualization strategies* with different maps on a smartphone and gathered feedback from
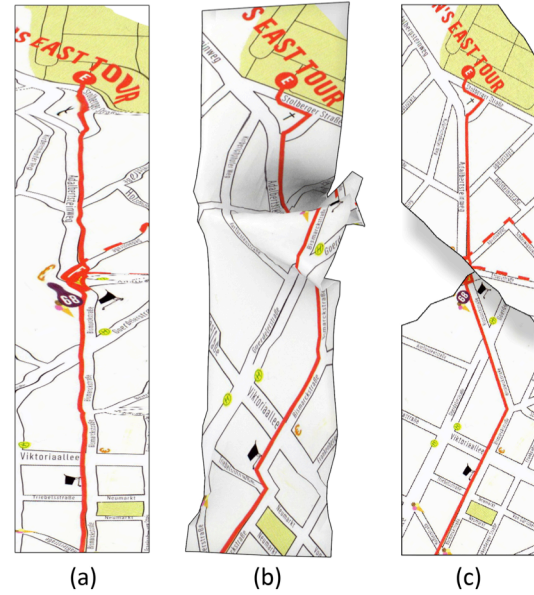
Figure 2: Comparison of different visualization strategies. The same route was converted using the transmogrification tool by Brosz [6] (a), an approach where we simulate that the map was printed on a piece of fabric and then was transformed into a stripe (b) and an approach in which we simulate that the map was printed on paper and then cut and reassembled (c).

different users. Each visualization strategy was tested on a smartwatch using various maps. We also presented the initial designs of the stripes to several experts (from the field of cartography as well as UI designers) and asked them to rate the design and discuss its pros and cons.

All users ranked option c best as it provided the best trade-off between information loss and a transformation easy to understand.

StripeMaps Algorithm

In a second step we developed an algorithm to perform the "paper" transformation strategy automatically for different maps. There are two main aspects for the transformation process: 1. "where" the cuts are performed and 2. "how" the cuts are performed. Cuts are not performed at every single waypoint because for zig-zag path segments this might result in a shattered stripe (referring to very small pieces in a row hard to interpret). Cuts are only performed if "not cutting" would result in a path beyond the left and right boundary of the stripe. The cuts are performed as follows: the paper is virtually cut along the angle bisector between the previous path segment and the following one. Then, the resulting pieces are rotated around the waypoint itself to achieve a straight path.

In detail, the transformation works as follows (Figure 3). Our algorithm accepts any map and a path, consisting of a start point S and an end point E with n points P in between. All ways between the points are described by straight lines (Figure 3a). The algorithm iterates through the path segments starting with the first three points S, P_1 and P_2.
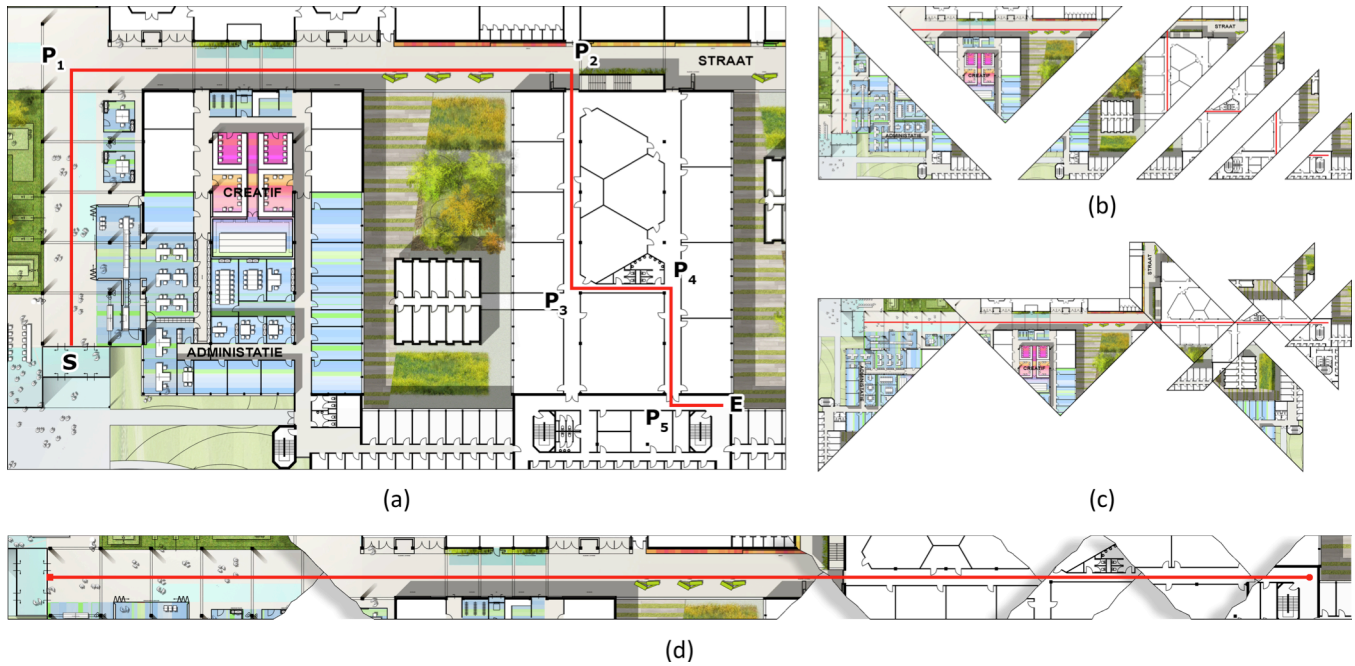
Figure 3: The "cutting and resembling" process from initial plan (a) over pieces (b) and merging them (c) to the final stripe (c).

Now the angle of the turn at P_1 is calculated. A virtual line (the angle bisector) is then drawn through P_1 and the segment of the image between S and P_1 is copied into the working memory. The next segment (P_1, P_2, and P_3) is then processed in the same way. To prevent corrupted pieces due to previous cuts, all the segments are cut out of the original map (Figure 3b). After the final segment (P_{n-1}, P_n and E) is processed, the algorithm arranges the pieces so that the path for every piece starts where the path of the previous piece ends. At the same time, all the pieces are merged together (Figure 3c). In the end, the resulting image is cropped and combined into one stripe (Figure 3d).

THE STRIPEMAPS APPLICATION

We have implemented the StripeMaps visualization and transformation approaches into our StripeMaps application for smartwatches. The StripeMaps application is targeted at the use case of pedestrian navigation, in particular indoor pedestrian navigation (although we also show how it can be extended to outdoor navigation as well in Study 2 below). More specifically, in the current version of the StripeMaps, we have used the following as our motivating use case: when a StripeMaps user enters a building whose floor plan has been loaded into the application, s/he receives a notification on her/his smartphone that StripeMaps is available to help navigating through the building. After selecting a target location using the smartphone (which displays the floor plan and a list of rooms to aid this process), a map stripe is generated and is pushed to the user's smartwatch for navigation. The user then uses the map stripe to get to her/his destination.

Below, we describe how we have implemented each step of this process and provide a high-level overview of the current version of the StripeMaps application.

System Overview

The StripeMaps application consists of three main parts. The central component is a smartwatch app to display the map and to support navigation by the user. A companion app, running on a smartphone, is used to display the initial floor plan and to perform the destination selection. The smartphone app also serves as a bridge between the smartwatch and the StripeMaps server. The server stores the map and list of possible destinations. When the user makes a route request from the smartphone application, the server performs the map transformation described above and delivers the StripeMap to the user's smartwatch (via the smartphone companion app).

When the user starts the companion app on the mobile device, the app downloads a list of available maps (e.g. different buildings on a campus) from the server. Then, the user selects the map s/he needs. The user's choice is sent back to the server. In the future, both can be done automatically e.g. by using GPS or Bluetooth beacons. The server answers with a thumbnail view of the selected map and a list of available destinations, e.g. rooms. After the user has selected the target location, a path from the starting point to the target location is sent to the server, which creates and returns the StripeMap. This approach could also allow the user to define a personalized path (e.g. to design one's own route through a museum), although this is not yet implemented. The server also calculates the geomagnetic orientation for all path segments. This allows extending StripeMaps with an orientation indicator: it shows the direction for the path segment the participant is currently looking at (for details please see user study section). After the mobile device has received the StripeMap and the directions, both can be pushed to the smartwatch app.
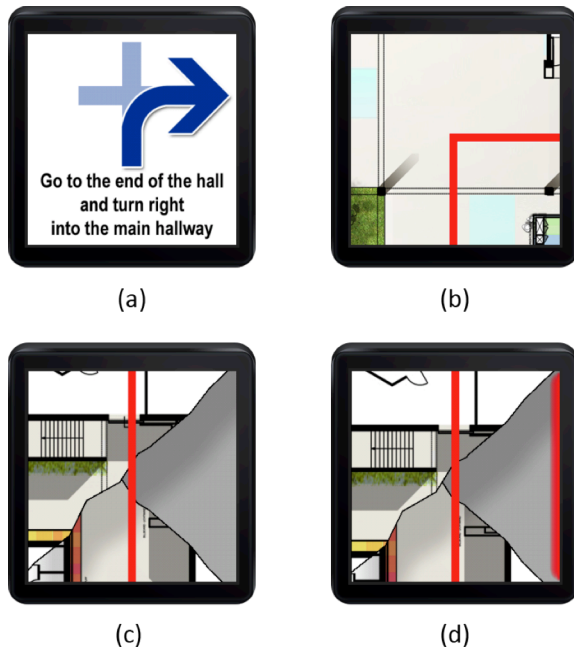
(a)   (b)

(c)   (d)

Figure 4: The four different conditions of the first user study. Turn-by-turn based navigation (TBT) (a), a 2D map that can be panned (2D) (b), the StripeMap application (SM) (c) and StripeMap with additional orientation indicator (SM+O) (d).

For the implementation we used an Android Wear smartwatch (Samsung Gear Live), an Android mobile device (Google LG Nexus 4), and a Java-based server application (running on a desktop PC). The communication between mobile device and smartwatch is realized using Bluetooth via the Android Wear platform, the communication between smartphone and server is based on TCP/IP, so they are enabled to communicate both via Wi-Fi and mobile network.

EVALUATION

We performed two different user studies to test the utility and effectiveness of the StripeMaps smartwatch application for pedestrian navigation. The first study was focused on comparing the StripeMaps application against commonly used navigation techniques implemented on smartwatches in a controlled indoor environment. Specifically, we used standard turn-by-turn navigation and traditional mobile map navigation as baselines. The second study was designed to explore the use of the StripeMaps application "in the wild".

Study 1

The first study was conducted in the main university building on the campus of Hasselt University, Belgium.

Participants & Apparatus

We recruited 16 participants (8 males and 8 females) with an average age of 19.4 years. All of them were unfamiliar with the university building, as they had just started their studies at the university in the week of the experiment. None of our participants had used a smartwatch before. The user study was performed on a Samsung Gear Live as described in the implementation section.

Figure 5: A user testing the StripeMaps condition (SM).

Task & Procedure

The participants were introduced to the experiment and instructed to find four different offices and labs in the university building. We explained the navigation task to them but not the StripeMaps visualization approach. The maps used in the study were already pre-loaded onto the smartwatch by the experimenter, so the participants did not need to use the companion smartphone app (e.g. to enter the destination or to push the StripeMap to the smartwatch). The four navigation tasks had similar lengths and similar characteristics (e.g. # turns and complexity). All participants performed the test under all of the following conditions (within-subject design). The orders of the four conditions and of the routes were counterbalanced:

1) Turn-by-turn navigation instructions (TBT)

2) 2D Map (2D)

3) *StripeMaps* (SM)

4) *StripeMaps* with orientation indicator (SM+O)

The different conditions are illustrated in Figure 4.

The instructions for the turn-by-turn condition (TBT) were created independently by three long-term members of the university community. They worked as a group to iteratively merge their introductions together. The university community members were instructed to use best practices in turn-by-turn navigation, e.g. by referencing landmarks rather than distances between turns [18].

In the second condition (2D), the participants could pan through a typical 2D map of the building with the highlighted route in red but were not allowed to zoom in/out. Map interaction on smartwatches is highly problematic. Apple itself warns developers in their Watch SDK documentation to not let maps be any bigger than the watch display so the user can see it all without having to pan and zoom at all[4]. Kerber et al. [13] have identified similar problems. While panning is required for 2D map navigation without positioning, there is no need for zooming. Additionally, without zooming the map was at the
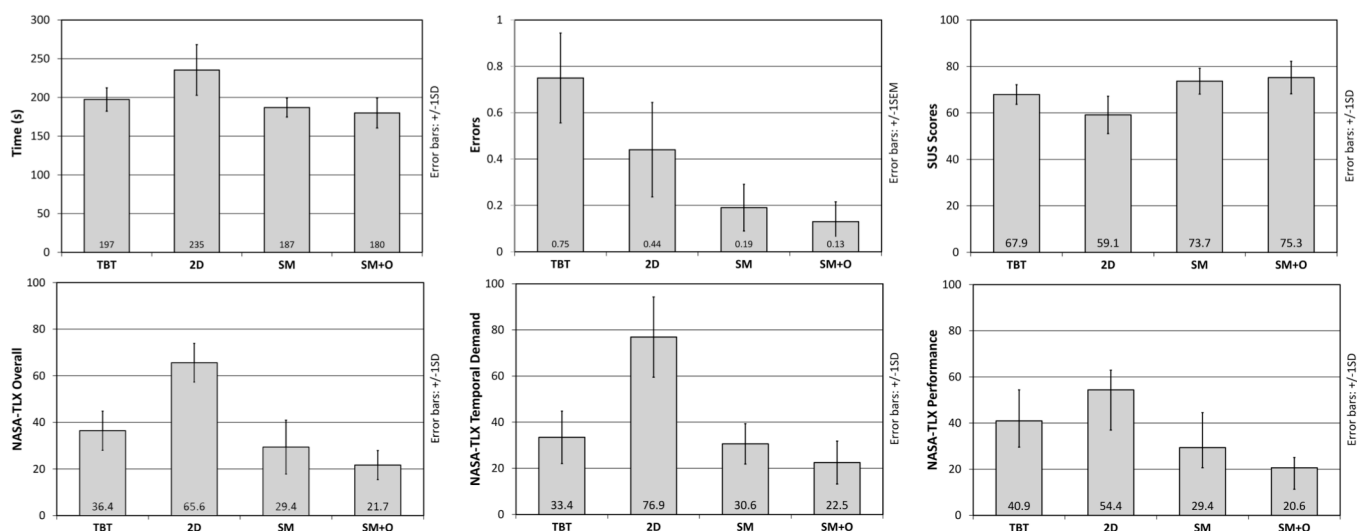

Figure 6: Overview of the results of the first study: mean times the participants needed to complete a single route, mean navigation errors per route, mean SUS scores, mean NASA-TLX overall values, and mean NASA-TLX values for temporal demand and performance (from left to right, from top to bottom).

same scale as in the SM and SM+O conditions, so the users could see the same amount of the map in all three conditions on the screen.

In the third condition (SM), the participants used the StripeMaps application as described above. In the fourth condition (SM+O), an orientation indicator was added to the StripeMaps application. The indicator had two functions. First, using the smartwatch's built-in compass it shows the direction for the path segment the participant is currently looking at. Second, the StripeMaps application in SM+O does not allow the user to scroll "down" the route as long as s/he has not turned in the correct direction. More specifically, when a participant comes to a cut in the StripeMap, before s/he can further scroll along the route, s/he needs to turn in the direction shown by the indicator. Once s/he does, the StripeMaps application allows her/him to scroll further.

As participants walked the route, an experimenter shadowed them, collecting timing information and counting the number of navigation errors they made. An error was tallied when a participant took a wrong turn without noticing within five meters. After five meters, the participant was told of their error and sent in the correct direction.

We used the System Usability Scale (SUS) [5] to measure the perceived usability and the NASA-TLX [9] to measure the perceived workload in all four conditions. Both questionnaires were filled in after each navigation task. The total time taken by each participant for the whole study was about 50 to 60 minutes. Participants were encouraged to think aloud and to ask questions if necessary. Noteworthy incidents were recorded in writing. A semi-structured interview was conducted with each participant afterwards.

Results and Analysis

All participants were able to complete all the tasks. The participants took about 200 seconds per route on average and made an average of 0.38 errors per route. Results are summarized in Figure 6.

The SM+O condition performed best with 180 seconds on average with a very low error rate of 0.13 (maximum of 1 error on each route). Just two of 16 participants took a wrong turn. The second best was the SM condition with an average completion time of 187 seconds and an error rate of 0.19 (max 2 errors). In the TBT condition, participants took about 197 seconds on average, and made about 0.75 errors on average per route (maximum of 3 errors). The TBT condition resulted in the most errors. The participants took the most time with the 2D map condition (235 seconds on average and 0.44 errors).

It is noteworthy that participants spent most of the additional time in the 2D condition, switching between the interaction on the smartwatch and then navigating through the building, e.g. 12 out of 16 participants stopped at every decision point to interact with the map, whereas in the SM and SM+O condition just 2 out of 16 participants stopped to interact with the smartwatch.

A one-way ANOVA for repeated measures revealed significant differences in time ($F_{3,45} = 21.1, p < .001$). Sidak-corrected pairwise analysis confirmed significant differences between TBT and the 2D condition ($p < .01$), SM and the 2D condition ($p < .01$), SM+O and TBT ($p < .05$) as well as SM+O and the 2D condition ($p < .001$). Overall, with regard to time to complete a route, both StripeMaps variants outperform a 2D map. Moreover, StripeMaps with the orientation indicator is significantly better than turn-by-turn navigation instructions, making it the fastest of all four conditions.

For error, a non-parametric Friedman test revealed significant differences ($\chi^2 = 10, p = .02$). Bonferroni-corrected (significance level of .0083) pairwise Wilcoxon tests confirmed that SM+O resulted in significantly fewer errors than TBT ($p = .004$). In other words, SM+O results in fewer navigation errors than turn-by-turn directions. StripeMaps without an orientation indicator also resulted in fewer errors, but it was not significant with Bonferroni correction. There were no significant differences between the two StripeMaps conditions and the standard 2D map condition.

Mean SUS scores can be considered as good for both StripeMaps conditions (SM = 73.7 and SM+O = 75.3), followed by an average score of 67.9 for TBT and the 2D condition (59.1). A one-way ANOVA for repeated measures ($F_{3,45} = 22.5, p < .001$) and post-hoc Sidak-corrected pairwise analysis confirmed statistical significance between all four conditions except between SM and SM+O: between TBT and the 2D condition ($p < .01$), SM and the 2D condition ($p < .001$), SM and TBT ($p = .045$), SM+O and the 2D ($p < .001$) condition as well as between SM+O and TBT ($p < .01$). Overall, regarding the perceived usability, the StripeMaps concept in both variants outperforms both turn-by-turn navigation instructions and traditional 2D maps.

The mean overall values for the NASA-TLX (normalized between 0 and 100) are low for both StripeMaps conditions (SM = 29.4 and SM+O = 21.7), which means that the workload is also low. While this is also true for the average value of 36.4 for TBT, the workload for the 2D condition was high (65.6). Again, a one-way ANOVA for repeated measures reveals significant differences ($F_{3,45} = 85.4, p < .001$). Sidak-corrected pairwise analysis confirmed significant differences between TBT and the 2D condition ($p < .001$), SM and 2D ($p < .001$), SM+O and 2D ($p < .001$) as well as SM+O and TBT ($p < .001$). That means, that while SM achieved a low workload comparable to TBT although it shows a more complex visualization (map), SM+O even outperformed TBT.

Noteworthy are also the results for the subscales of temporal demand and performance. The temporal demand for the 2D condition (76.9) is much higher than for all the others (TBT = 33.4, SM = 30.6 and SM+O = 22.5). A one-way ANOVA for repeated measures reveals significant differences ($F_{3,45} = 68.9, p < .001$). Sidak-corrected pairwise analysis confirmed significant difference between TBT and the 2D condition ($p < .001$), SM and 2D ($p < .001$), SM+O and 2D ($p < .001$) as well as SM+O and TBT ($p = .018$). The participants noticed that they needed more time to complete the navigation tasks in the 2D condition than in the other ones. Also the perceived performance was higher for the 2D condition (54.4) than for TBT (40.9), SM (29.4) and SM+O (20.6). Again, one-way ANOVA for repeated measures ($F_{3,45} = 26.6, p < .001$) and post-hoc Sidak-corrected pairwise analysis confirmed

significant differences between TBT and 2D ($p = .015$), SM and 2D ($p < .001$), SM+O and 2D ($p < .001$) as well as SM+O and TBT ($p < .001$).

In the semi-structured interviews, all participants perceived the use of the StripeMaps smartwatch app positively. They commented that the StripeMaps application "makes life so much easier" (P5) and "that it provides a good trade-off between clear instructions and a full a map" (P12). P3 commented on the 2D map condition that "it was incredible hard not to get lost on the map – with the SM I could just flick through the map with a fast swipe. With the map I was always using the whole screen space with my fingers and got lost over and over again. It was so frustrating".

At the end of the test runs, after the participants had navigated using all four conditions, we asked them to rank the interfaces. All participants favored one of the two StripeMaps variants. Three of the 16 participants ranked SM first, while 13 ranked SM+O first. Those that did not rank SM+O first ranked it second. All of the participants except one ranked the standard mobile map 2D condition last.

Study 2
In order to gain a more qualitative understanding of users' interaction with StripeMaps and to see how it is used "in the wild" rather than in a controlled environment, we conducted a second user study. The study took place in Aachen, Germany. During the year 2014, the city celebrated a major anniversary of the death of an important historical figure and former resident, drawing many tourists into the city.

We drew our participants from this population of tourists. We used the official tourist map of the city and turned the tours shown on the paper map into StripeMaps (using the SM condition from the first study). Again, the maps were pre-loaded onto the smartwatch by the experimenter. We then approached randomly selected tourists close to different sights along the tours and asked if they would like to try out our research prototype.

Procedure
Eight participants (4 female, 4 male; from groups of 2-8 persons) with a mean age of 38.5 years took part in the study. The study was conducted across two days in summer 2014 with good weather conditions of about 24°C and sunshine. After the participants agreed to take part in the user study, we introduced them to the application. The experimenter then joined the participants while they were continuing their activity (visiting the city). After 20 minutes, we asked the participants to provide us feedback on the app and fill in a SUS questionnaire. This was followed by an informal interview with the participants.

Results
Overall, the StripeMaps application received a SUS score of 79 on average, which is quite similar to the first study's 73.7. Participants also offered positive qualitative feedback

P5[5] mentioned that "the main advantage of this thing is that I do not have to pull out the map or my mobile out of my pocket again and again. I would like to have this on my trip to Barcelona, as it makes me feel more secure." (referring to her worries about getting her smartphone stolen). She continued that "the map does not distract a lot, it is simple and nice – you can just scroll through. I can also imagine this as a map for my next marathon". Several participants (P3, P6, P7, P8) commented that it was also fun to use. P6: "It's fun – the city tour is a simple run on a stripe. This makes the tour very easy. I am often confused by complex city structures. Here the structure is broken down into a stripe". She also expressed a desire to print out all StripeMaps of tours she has done to collect them.

P8 missed that the StripeMaps application did not make use of the GPS, he said that "If you would add the blue GPS dot, then it would be perfect, but still this is great. GPS still has some problems in some cities. A simple modification would also be if the stripe scrolls automatically with your walking speed. Nevertheless, this would work better than Google Maps shown on the Apple watch, this is cool!"

DISCUSSION
Smartphones are not an ideal platform for pedestrian navigation. Weaknesses include the frequent need for two-handed interaction and having to take the device out of a purse or pocket in order to view a route. This interaction overhead creates distractions, with pedestrians potentially overlooking risks of oncoming traffic or obstacles in their way. As reported in the second study, this could also increase the user's sense of security. Smartglasses are the most unobtrusive interface and certainly solve some problems; but while supporting always-visible navigation instructions, they lack direct forms of interaction and currently can incur some social costs [10].

Smartwatches are a compromise: they potentially incur fewer social costs, provide direct interaction and are always at hand. The problem, however, is that their screens are unprecedentedly small for interactive smart devices. In our studies, we have provided evidence that StripeMaps can help to overcome this obstacle in the use of smartwatches in pedestrian navigation. Specifically, we have shown that StripeMaps results in faster route completion with fewer errors than other smartwatch cartographic approaches. This includes standard mobile map cartographic approaches like those being employed on the Apple Watch. StripeMaps also received positive qualitative feedback in our user studies, with participants expressing explicit enjoyment in the reduction in spatial complexity that was the goal of the StripeMaps approach.

As noted above, in addition to introducing StripeMaps and demonstrating its effectiveness, this work also represents

the first investigation of cartographic approaches for smartwatches. In this respect, our findings have implications beyond StripeMaps. Namely, we saw evidence that while turn-by-turn directions result in faster navigation than standard mobile map cartography displayed on a smartwatch, turn-by-turn directions also result in significantly more errors. Additionally, in our user study, if a participant made an error, the experimenter corrected her/him after five meters. It is possible that given its error rate, turn-by-turn directions would result in much longer route completion times "in the wild", perhaps even longer than standard mobile maps. Even though the standard maps could not reach the same performance and ratings as StripeMaps, it appears they are still better than turn-by-turn instructions for pedestrian navigation on smartwatches.

While we compared StripeMaps against current smartwatch map approaches, we did not compare StripeMaps against the traditional "turn-by-turn"-style 3D display that is included as a feature in most smartphone map apps (as well as in traditional GPS devices). Full-featured turn-by-turn displays are not yet available on smartwatches, likely due to computing and screen size constraints, and as such, StripeMaps' performance relative to these displays is not known. However, turn-by-turn 3D displays require highly accurate position information, and this information is not available in indoor environments. StripeMaps does not require position information, making it suitable for indoor navigation, which is a critical use case for pedestrian navigation (e.g. campus buildings, malls, airports). If and when turn-by-turn displays become available for smartwatches, we hope to be able to compare StripeMaps against them for *outdoor* pedestrian navigation.

It is also important to reiterate that StripeMaps is explicitly dedicated to map *navigation* rather than *orientation* (i.e. "getting one's bearings"), which is the other primary use case for reference maps (the family of maps to which mobile maps belong). There is a long tradition of developing technologies specifically for navigation (e.g. print directions) as we did for StripeMaps, and new techniques will have to be developed to aid in smartwatch map orientation. This is a subject of future work for us, and it is likely that similarly non-traditional approaches will be necessary, as orientation is also quite dependent on screen size. Another limitation is that the current version of the StripeMaps application turns rasterized images into stripes. While pixel-based maps are widespread and simple to use, a serious drawback are rotated labels (e.g. street names or symbols of landmarks), which can become unreadable or unrecognizable. In the future we are planning to develop a system providing vector-based maps.

Finally, this paper has provided evidence that linearization has benefits in cartography, just as it does in mobile web design. The design space of linearization techniques is large, and further research should identify and compare different alternatives.

---

[5] Please note that we had different user groups in the first and second study. Participant numbers in this section refer to study 2.

REFERENCES

1. Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., & Pinkerton, M. (1997). Cyberguide: A Mobile Context-aware Tour Guide. Wireless Networks, 3(5), pp. 421-433.

2. Alvina, J., Appert, C., Chapuis, O., & Pietriga, E. (2014). Routelens: Easy Route Following for Map Applications. In Proc. of the International Working Conference on Advanced Visual Interfaces.

3. Ashbrook, D., Lyons, K., & Starner, T. (2008). An Investigation into Round Touchscreen Wristwatch Interaction. In Proc. of MobileHCI 2008 (pp. 311-314). ACM.

4. Baus, J., Cheverst, K., & Kray, C. (2005). A Survey of Map-based Mobile Guides. In Map-based Mobile Services (pp. 193-209). Springer.

5. Brooke, J. (1996). SUS-A Quick and Dirty Usability Scale. Usability Evaluation in Industry, 189.

6. Brosz, J., Nacenta, M. A., Pusch, R., Carpendale, S., & Hurter, C. (2013). Transmogrification: Causal Manipulation of Visualizations. In Proc. of UIST 2013. ACM.

7. Chen, X. A., Grossman, T., Wigdor, D. J., & Fitzmaurice, G. (2014). Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. In Proc. of CHI 2014 (pp. 159-168). ACM.

8. Cheverst, K., Davies, N., Mitchell, K., Friday, A., & Efstratiou, C. (2000). Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. In Proc. of CHI 2000 (pp. 17-24). ACM.

9. Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. Advances in Psychology, 52 (pp. 139-183).

10. Hong, J. (2013). Considering Privacy Issues in the Context of Google Glass. In Communications of ACM, 56(11) (pp. 10-11).

11. Huang, H., & Gartner, G. (2010). A Survey of Mobile Indoor Navigation Systems (pp. 305-319). Springer.

12. Hwang, Y., Kim, J., & Seo, E. (2003). Structure-aware Web Transcoding for Mobile Devices. Internet Computing, IEEE, 7(5), (pp. 14-21).

13. Kerber, F., Krüger, A., & Löchtefeld, M. (2014). Investigating the Effectiveness of Peephole Interaction for Smartwatches in a Map Navigation Task. In Proc. of MobileHCI 2014 (pp. 291-294). ACM.

14. Kim, J., He, J., Lyons, K., & Starner, T. (2007). The Gesture Watch: A Wireless Contact-free Gesture based Wrist Interface. In Proc. of Wearable Computers (pp. 15-22). IEEE.

15. Kray, C., Elting, C., Laakso, K., & Coors, V. (2003). Presenting Route Instructions on Mobile Devices. In Proc. of IUI 2003 (pp. 117-124). ACM.

16. Malaka, R., & Zipf, A. (2000). Deep Map: Challenging IT Research in the Framework of a Tourist Information System (pp. 15-27). Springer Vienna.

17. Marcotte, E. (2011). Responsive Web Design. A Book Apart.

18. May, A. J., Ross, T., Bayer, S. H., & Tarkiainen, M. J. (2003). Pedestrian Navigation Aids: Information Requirements and Design Implications. Personal and Ubiquitous Computing, 7(6), pp. 331-338.

19. MacEachren, A. M. (2004). How maps work: representation, visualization, and design. Guilford Press.

20. McGookin, D. K., & Brewster, S. A. (2013). Investigating and Supporting Undirected Navigation for Runners. In CHI 2014 Extended Abstracts (pp. 1395-1400). ACM.

21. Möller, A., Kranz, M., Diewald, S., Roalter, L., Huitl, R., Stockinger, T., Koelle, M., & Lindemann, P. A. (2014). Experimental Evaluation of User Interfaces for Visual Indoor Navigation. In Proc. of CHI 2014 (pp. 3607-3616). ACM.

22. Mulloni, A., Seichter, H., & Schmalstieg, D. (2011). Handheld Augmented Reality Indoor Navigation with Activity-based Instructions. In Proc. of MobileHCI 2011 (pp. 211-220). ACM.

23. Nurminen, A. (2006). m-LOMA-a mobile 3D city map. In Proc. of 3D web technology 2006 (pp. 7-18). ACM.

24. Oakley, I., & Lee, D. (2014). Interaction on the Edge: Offset Sensing for Small Devices. In Proc. of CHI 2014 (pp. 169-178). ACM.

25. Pielot, M., Henze, N., & Boll, S. (2009). Supporting Map-based Wayfinding with Tactile Cues. In Proc. of MobileHCI 2009 (p. 23-33). ACM.

26. Puikkonen, A., Sarjanoja, A. H., Haveri, M., Huhtala, J., & Häkkilä, J. (2009). Towards Designing Better Maps for Indoor Navigation: Experiences from a Case Study. In Proc. of MUM 2009. ACM.

27. Reilly, D. F., Inkpen, K. M., & Watters, C. R. (2009, September). Getting the Picture: Examining how Feedback and Layout Impact Mobile Device Interaction with Maps on Physical Media. In Proc. of ISWC'09 (pp. 55-62). IEEE.

28. Rekimoto, J. (1996). Tilting Operations for Small Screen Interfaces. In Proc. of UIST 1996. ACM.

29. Sarkar, M., Snibbe, S. S., Tversky, O. J., & Reiss, S. P. (1993). Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens. In Proc. of UIST 1993 (pp. 81-91). ACM.

30. Schöning, J., Hecht, B., & Kuhn, W. (2014). Informing Online and Mobile Map Design with the Collective Wisdom of Cartographers. In Proc. of DIS 2014. ACM.

31. Seager, W., & Fraser, D. S. (2007). Comparing Physical, Automatic and Manual Map Rotation for Pedestrian Navigation. In Proc. of CHI 2007 (pp. 767-776). ACM.

32. Siek, K. A., Rogers, Y., & Connelly, K. H. (2005). Fat finger worries: how older and younger users physically interact with PDAs. In Human-Computer Interaction-INTERACT 2005 (pp. 267-280). Springer Berlin Heidelberg.

33. Slocum, T. A., McMaster, R. B., Kessler, F. C., & Howard, H. H. (2009). Thematic Cartography and Geovisualisation. Prentice Hall.

34. Wenig, D., & Malaka, R. (2010). Interaction with Combinations of Maps and Images for Pedestrian Navigation and Virtual exploration. In Proc. of MobileHCI 2010 (pp. 377-378). ACM.

35. Xiao, R., Laput, G., & Harrison, C. (2014). Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click. In Proc. of CHI 2014 (pp. 193-196). ACM.

36. Zhang, D. (2007). Web content adaptation for mobile handheld devices. Communications of the ACM, 50(2), (pp. 75-79).