

# Combining user logging with eye tracking for interactive and dynamic applications

Kristien Ooms · Arzu Coltekin · Philippe De Maeyer · Lien Dupont · Sara Fabrikant · Annelies Incoul · Matthias Kuhn · Hendrik Slabbinck · Pieter Vansteenkiste · Lise Van der Haegen

Published online: 10 December 2014  
© Psychonomic Society, Inc. 2014

**Abstract** User evaluations of interactive and dynamic applications face various challenges related to the active nature of these displays. For example, users can often zoom and pan on digital products, and these interactions cause changes in the extent and/or level of detail of the stimulus. Therefore, in eye tracking studies, when a user's gaze is at a particular screen position (gaze position) over a period of time, the information contained in this particular position may have changed. Such digital activities are commonplace in modern life, yet it has been difficult to automatically compare the changing information at the viewed position, especially across many participants. Existing solutions typically involve tedious and time-consuming manual work. In this article, we propose a methodology that can overcome this problem. By combining *eye tracking* with *user logging* (mouse and keyboard actions) with cartographic products, we are able to accurately reference screen coordinates to geographic coordinates. This referencing approach allows researchers to know which geographic object (location or attribute) corresponds to the gaze coordinates at all times. We tested the proposed approach through two case studies, and discuss the advantages and

disadvantages of the applied methodology. Furthermore, the applicability of the proposed approach is discussed with respect to other fields of research that use eye tracking—namely, marketing, sports and movement sciences, and experimental psychology. From these case studies and discussions, we conclude that combining eye tracking and user-logging data is an essential step forward in efficiently studying user behavior with interactive and static stimuli in multiple research fields.

**Keywords** Eye tracking · Interactivity · Georeferencing

Eye tracking has proven to be a helpful technique in user research, especially when a visual element needs to be evaluated. By using eye tracking data, researchers can discover how long and how often a user looks at a particular area of interest, as well as the length and speed of the eye movements (Duchowski, 2007; Holmqvist et al., 2011). The position of the gaze (also termed the *point of regard*, or POR) is typically expressed using screen coordinates in pixels. From these basic screen coordinate measurements, various gaze metrics are derived in relation to what is displayed, such as the fixation duration (how long), fixation count (how often), and various scan-path characteristics (e.g., the length and speed of eye movements). The technique has been applied in a multitude of research fields, including software engineering, industrial engineering (e.g., driving, aviation), marketing (e.g., ad placement, webpages, product label design), psychology (e.g., reading, scene perception, visual search), cartography (e.g., map reading, orientation, way finding), sports and movement sciences (e.g., tactile decision making), landscape perception and design, and so forth (e.g., Allopenna, Magnuson, & Tanenhaus, 1998; Brodersen, Andersen, & Weber, 2001; Duchowski, 2007; Dupont, Antrop, & Van Eetvelde, 2013a; Goldberg, Stimson, Lewenstein, Scott, & Wichansky,

---

K. Ooms (✉) · P. De Maeyer · L. Dupont · A. Incoul  
Department of Geography, Ghent University, Ghent, Belgium  
e-mail: kristien.ooms@ugent.be

A. Coltekin · S. Fabrikant · M. Kuhn  
Department of Geography, University of Zurich, Zurich, Switzerland

H. Slabbinck  
Department of Marketing, Ghent University, Ghent, Belgium

P. Vansteenkiste  
Department of Sports and Movement Sciences, Ghent University,  
Ghent, Belgium

L. Van der Haegen  
Department of Experimental Psychology, Ghent University,  
Ghent, Belgium

2002; Jacob & Karn, 2003; Nivala, Sarjakoski, Jakobsson, & Kaasinen, 2001; Pieters, 2008; Poole & Ball, 2006; Rayner, 1998, 2009; Recarte & Nunes, 2000; Wedel & Pieters, 2006)

During the last century, visual stimuli have evolved dramatically, from analog and static (e.g., Buswell, 1935; Fitts, Jones, & Milton, 1950) to digital and interactive (both offline and online). Thus, it is important to gain a better understanding of how users perceive, process, and react to interactive visual stimuli. Due to user interactions and/or animations, changes in a visual stimulus—such as changes in the color or location of an object, the (dis)appearance of an object, and so forth—will occur. Research on change blindness, for example, has shown that some of these changes, although clearly visible, may go unnoticed by users (Garlandini & Fabrikant, 2009; Rensink, 2002; Simons & Ambinder, 2005). This evolution of visual stimuli and the associated problems with respect to eye tracking studies can be well illustrated in the field of cartography, wherein maps are the focus of interest.

### Cartography: A special interactive case

Since cartographic products are visual in nature, eye tracking has been helpful in studying map users' cognitive processes while working with different map types and related products. Early studies tested static maps, initially on paper (Castner & Eastman, 1984, 1985; Dobson, 1977; Steinke, 1979, 1987), but later also on digital media (Brodersen et al., 2001). In the last decades, psychological research on the cognitive processes linked with visual search has received much attention, thus resulting in new and more detailed theories regarding cognitive cartography (e.g., Harrower, 2007; Hegarty, Canham, & Fabrikant, 2010; MacEachren, 1995; Slocum et al., 2001). Perhaps as a result of this, a renewed interest in the use of eye tracking in cartographic studies has been observed (Coltekin, Fabrikant, & Lacayo, 2010; Coltekin, Heil, Garlandini, & Fabrikant, 2009; Dong, Liao, Roth, & Wang, 2014; Fabrikant & Lobben, 2009; Fabrikant, Rebich-Hespanha, Andrienko, Andrienko, & Montello, 2008; Incoul, Ooms, & De Maeyer, 2015; Popelka & Brychtova, 2013).

Recent digital cartographic products—both online and offline—are typically linked with a number of interactive tools that overcome one of their most important drawbacks, in comparison to paper maps—limited screen size (e.g., Brewster, 2002; Kraak & Brown, 2001; Peterson, 2003). According to Shneiderman (1992), users of information visualizations (such as maps) typically want to have an *overview* of the data first, to select the appropriate region (*zoom-and-filter*), and then to request its details (*details-on-demand*). In accordance with Shneiderman's observation, zooming and panning tools can be found on nearly all digital cartographic products today, thus allowing for iteration between overviews

and detail views (Luebbering, Carstensen, Campbell, & Grossman, 2008; Roth, 2011).

User studies that have incorporated the interactive nature of digital cartographic products, however, are rare (Coltekin et al., 2009; Russo et al. 2014). Typically, the interactive nature of maps is approximated; for example, the maps are implemented as a collection of static images or videos. These videos simulate a certain user action with the same start time, duration, and direction—for example, the simulation of a pan operation in Ooms, De Maeyer, Fack, Van Assche, and Witlox (2012). This approximation facilitates the processing, comparing, and analyzing of the obtained data. However, it also means that the users cannot freely interact with the map. In other words, the users cannot choose when to interact; select the panning distance; identify which zoom level they deem most appropriate for a specific task, with respect to increasing or decreasing the level of detail; choose to tilt or rotate the display; or decide whether to use a search box. Ideally, under experimental conditions, participants should execute a task on the interactive map as they would *normally* do so, without restrictions on their behaviors or on the interactivity levels of the tested display. Testing the users in situations that more closely mimic their natural work routines would increase the ecological validity of the experiment.

On the other hand, an ecologically valid approach with interactive maps would introduce severe challenges to the internal validity of the experiments and create challenges with respect to analyzing the data. For example, it is clear that each participant would start her or his interaction (e.g., pan operation) with the map at a different timestamp, which would complicate the analyses among participants. Further consideration of the panning example indicates that the panning distance (the distance between the mouse-key-down and mouse-key-up actions—i.e., between pressing and releasing the left mouse button) would vary with each interaction, such as the direction of the pan operation. In other words, after a panning operation, the (pixel or screen) coordinates in the upper left corner of the screen would remain fixed (e.g., at 0, 0), though another geographic region was being visualized.

Partly because of the challenges of evaluating dynamic stimuli, we still know very little about how end users actually read, interpret, and process interactive maps and other similar interactive applications. Various studies have shown that interaction tools surrounding the maps in digital environments may hinder effective and efficient information extraction, and thus affect the usability of the systems (Fabrikant & Lobben, 2009; MacEachren & Kraak, 2001; Montello, 2009). To this day, many (design) issues related to dynamic and interactive maps are not yet well understood (Cartwright, 2012; van Elzakker & Griffin, 2013; Virrantaus, Fairbairn, & Kraak, 2009). How the changing map display affects the users' cognitive processes during a spatial task remains one of these challenges.

However, when working with cartographic products we have an advantage, since every point in the map is defined by its geographic coordinates. Ideally, eye tracking data in screen coordinates could be transformed to geographic coordinates by using a fairly simple referencing process. That is, *georeferencing the eye tracking data* would overcome the aforementioned problems regarding the evaluation of dynamic map stimuli, thus potentially allowing for more efficient analyses and comparisons than with current techniques.

In this article, state-of-the-art methods and techniques that try to deal with the dynamic and interactive nature of stimuli in combination with eye tracking studies will be presented, including the drawbacks of such techniques. Next, we will describe various approaches to transform the registered screen coordinates to geographic coordinates. One of the most complete solutions will then be tested in a number of case studies. Furthermore, the applicability of the concept in other research fields—namely, experimental psychology, landscape research, sports and movement sciences, and marketing—will be discussed.

### Eye tracking and dynamic stimuli: Existing solutions

Over the years, the vendors of eye tracking software and researchers in the field have developed methods and techniques that attempt to address the dynamic and interactive nature of digital stimuli, at the level of both data acquisition and data analyses. Some of these are a consequence of evolutions in the eye tracking systems themselves, such as the increasing use of mobile eye trackers (e.g., SMI Eye Tracking Glasses<sup>1</sup>; Tobii Glasses Eye Tracker<sup>2</sup>) (Kiefer, Giannopoulos, & Raubal, 2014; Reimer & Sodhi, 2006). With such eye trackers, users can walk around freely while their gaze position, as well as a video of their visual field, is being recorded. During analyses, the user's gaze position is typically overlaid onto this dynamic video, which differs for each user. A similar approach is often used in the recording and analyses of eye movement data on interactive and dynamic stimuli from static eye trackers, with all events on the screen being recorded in a video through screen capturing. However, all of the resulting videos overlaid with associated eye movement data have to be processed individually and, for the most part, manually, which is a very time-consuming and potentially subjective task. Accordingly, some solutions have been developed to facilitate this process.

One possible approach is the use of dynamic areas of interest (AOIs), which are currently implemented in the

analysis software of most eye tracking vendors, in place of the traditional static AOIs (Holmqvist et al., 2011). Dynamic AOIs can be defined on dynamic stimuli, such as videos, as the result of mobile eye tracking devices or screen recordings of interactive digital stimuli. As such, dynamic AOI define the position and size (e.g., bounding box) of an object of interest in the dynamic stimuli, and as a consequence, the dynamic AOI will change in position and size over time, thus following the object in the stimulus. Papenmeier and Huff (2010) developed an open-source tool to define dynamic AOIs based on a 3-D model of the visual scene. They also present an overview of existing approaches of dynamic AOIs in which they differentiate between online and offline AOIs. Nevertheless, these dynamic AOIs have an important drawback that drastically diminishes their usability in the case of interactive stimuli or mobile eye tracking devices: each dynamic AOI must be created manually. Some software packages facilitate this task as the manual definition and adjustment of the AOI is only required for a number of key frames, whereas the software then creates estimates for the frames in between. However, in the case of recordings from interactive stimuli, the resulting video would be different for every participant, which means that these dynamic AOIs would have to be drawn separately for each of participant. This results in extremely tedious and time-consuming manual work, which is not desirable.

Most software accompanying eye tracking systems (e.g., SMI Experiment Center, Tobii Studio, SR Research Experiment Builder) allow for the defining of certain parameters that should be recorded during the experiment, such as mouse actions. These mouse actions are a vital source of information because they are the triggers for the interactions that occur on the screen (Pirolli, Fu, Reeder, & Card, 2002; Reeder, Pirolli, & Card, 2001). Because the mouse actions and the eye tracking data would ideally be registered by the same system, no synchronization issues would arise. However, many commercial systems do not make a distinction between the mouse-down and the mouse-up actions, but rather they record only mouse clicks (i.e., mouse key press is recorded but user release of the key is not recorded). Consequently, mouse movements and dragging (moving the mouse while one of its keys is pressed) cannot be registered, and thus cannot be analyzed, which is essential for certain studies, such as the zooming functionality by drawing a rectangle or by panning. An exception on this is GazeTracker.<sup>3</sup> Various eye tracking vendors, such as Tobii<sup>4</sup> and SMI,<sup>5</sup> offer SDKs (software development kits) that can be used for creating custom solutions based on the existing software. Similarly, some others—such as Morae<sup>6</sup>—offer plugins for certain eye tracking

<sup>1</sup> [www.eyetracking-glasses.com/products/eye-tracking-glasses-2-wireless/technology/](http://www.eyetracking-glasses.com/products/eye-tracking-glasses-2-wireless/technology/)

<sup>2</sup> [www.tobii.com/en/eye-tracking-research/global/library/videos/tobii-glasses/tobii-glasses-2-eye-tracker/](http://www.tobii.com/en/eye-tracking-research/global/library/videos/tobii-glasses/tobii-glasses-2-eye-tracker/)

<sup>3</sup> [www.eyetellect.com/gazetracker/](http://www.eyetellect.com/gazetracker/)

<sup>4</sup> [www.tobii.com/](http://www.tobii.com/)

<sup>5</sup> [www.smivision.com/en.html](http://www.smivision.com/en.html)

<sup>6</sup> [www.techsmith.com/morae-plugins.html](http://www.techsmith.com/morae-plugins.html)

software (e.g., Tobii), thus allowing a detailed logging of mouse actions (clicks, movements, etc.) simultaneously with eye movements. A continuous sampling of mouse movements can be obtained and has been used in, for example, hand–eye coordination studies (Coltekin, Demsar, Brychtova, & Vandrol, 2014).

Based on recorded mouse actions (mouse clicks corresponding to user interactions), the analysis software from Tobii (Tobii Studio), for example, also allows segmenting the recorded screen videos such that every segment represents a time interval during which no interactions occur. However, when one wants to evaluate users' attentive behaviors across a large number of participants, the corresponding segments of all participants (e.g., when viewing the same image) must be manually determined, which is, again, a very time-consuming and tedious job. Furthermore, the latter solutions are vendor specific, meaning that to work with other types of eye tracking devices or data, the code would have to be adopted or rewritten.

SMI also offers a Video Data Aggregation Package as an aid in the analyses of eye movement data on dynamic stimuli. With this package, one can map fixations that originally occur on an object in the video to a reference image using SMI Semantic Gaze Mapping. All potentially interesting objects that are visible in the video should be present in the static reference image. The analyses of the eye movement data are conducted on the static reference image rather than on the dynamic video. However, an addition to being a vendor specific solution, all fixations must be mapped manually to the reference image, which is, again, a very time consuming and tedious job.

A platform-independent automated solution with finer mouse-logging behavior that included mouse-up and mouse-down actions would introduce a significant benefit to researchers and practitioners with respect to user experience. Especially a solution based on *open source software* would provide sufficient flexibility to adapt the 'standard' solution to the experimenter's needs. Using open source libraries removes dependencies on other (commercial or specialized) software as well as on vendor specific eye tracking hardware.

User logging is not a new methodology as it has been extensively used for many decades in User Centered Design (UCD) to gather quantitative data from end users who execute a certain task on a certain product (e.g., Hilbert & Redmiles, 2000; Ivory & Hearst, 2001; Paganelli & Paternò, 2002; e.g., Atterer, Wnuk, & Schmidt, 2006; Wengelin et al. 2009). Through user logging, we can discover, e.g., where users are clicking in an interface, how often certain button combinations are used, whether certain menu items can be found and when the user action occurs. These data provide insights about the usability of the evaluated product (Nielsen, 1993). Van Drunen, van den Broek, Spink, and Heffelaar (2009), for example, recorded user actions as an indication of user

workload while performing a Web-based task. However, the recorded mouse actions were not used in the analyses of the eye movements or screen captures (videos) recorded during the experiment, but rather the number of mouse movements were compared with the number of fixations (and other measurements). The position of the mouse movements was not considered in this research, however.

A promising solution are the tools developed by Reeder et al. (2001)—WebLogger and WebEyeMapper. With the online logging tool WebLogger, all user actions and other interesting events are logged and saved. These logs can be loaded into WebEyeMapper, along with the recorded eye movements, thus creating a reconstruction of the webpages the participant was viewing that includes the locations of his/her fixations (Pirolli et al., 2002). Although very promising, the disadvantage of this solution is that it is limited to online stimuli that can only be loaded in the Internet Explorer browser.

Users also perform mouse and keyboard actions when working with interactive cartographic products. This can include mouse actions such as clicking, dragging and scrolling, which reveal when and how the user is interacting with the digital map. Logging the mouse interactions might also provide vital data for linking the screen coordinates obtained by an eye tracker to the corresponding geographic coordinates, which will be further explored in this article. In the next sections, we focus on the selection and implementation of an appropriate method for logging mouse actions at a detailed level that can be combined with eye movement measurements. In the selection procedure, the focus is on the applicability of the methodology on the cartographic interactive problem because of its special geographic characteristics (i.e., potential georeferencing of eye movement data). Later on, the suitability of the selected methodology in other research fields is also discussed.

### **Solution for interactive cartographic products: Georeferencing eye movements**

Technical and conceptual description of potentially suitable user-logging approaches

In general, we distinguish between *online* (or *browser-based*) and *desktop-based* user-logging approaches. Online logging tools have a disadvantage in that only online applications or applications that work within a browser can be evaluated, whereas this is possible for both online and offline applications running with desktop-based tools. However, as most of the interactive cartographic products are available online, it is logical to use an *online* logging system, and accordingly, a number of potential promising solutions are also identified.



*Online solutions* For custom online solutions, most web-mapping providers (e.g., Google Maps, Bing Maps, etc.) provide application programming interfaces (APIs), which are also appropriate for our purposes. However, as we propose a “stand-alone” solution independent of other software, this approach is not optimal (e.g., Peterson, 2015; Roth & Ross, 2012). Furthermore, these APIs cannot be used in the case of neocartographic maps or mash-up maps. With these latter two types of maps, the base map (from an online map service such as Google Maps) is overlaid with one or multiple additional layer(s) of information from other sources: current position of airplanes, data from Twitter, precipitation data, and so forth (Cartwright, 2012; Das, van Elzakker, & Kraak, 2012; Haklay, Singleton, & Parker, 2008; Moseme & van Elzakker, 2012). The main advantage of APIs is that they allow the possibility to obtain access to the base map by requesting the associated geographic coordinates directly without having to calculate them and overlay the data; however, it is not possible to access third-party cartographic products or layers.

*Desktop-based solutions* In addition to the online logging tools, it is possible to create desktop-based logging tools. These tools are independent of any browser as the events are logged directly on the device of the participants. Various independent (desktop) programs exist whereby mouse actions can be recorded and replayed (e.g., ReMouse<sup>7</sup>). However, only a few of these programs offer the possibility to actually log the recorded data in an open readable format (such as a comma-separated or tab-separated file) or to distinguish between mouse-down and mouse-up actions, or to record the scroll wheel, thus again limiting their suitability for this application.

In addition to the APIs, it is possible to use existing tools or libraries, such as OpenLayers (Hazzard, 2011) as these provide nearly the same possibilities as the APIs in that they combine existing base maps with additional layers, but they also present the same limitations. For example, using these specific libraries, it is not possible to log user events on neocartographic products if they are not created within that specific library. This limitation makes the use of specific tools or libraries such as OpenLayers not ideal for the standalone solution that we propose.

When designing websites, HTML iFrames<sup>8</sup> are often used to incorporate a web page from a different source in the current one. It can be seen as a rectangle that links to another webpage through a URL. Attaching JavaScript to the main webpage, would, in theory, make it possible to log the users’ actions on this page. However, the mouse actions are not registered on the iFrame itself. To be able to access this data, the domains of the parent and child pages must be the same.

<sup>7</sup> [www.remouse.com/](http://www.remouse.com/)

<sup>8</sup> [www.w3schools.com/tags/tag\\_iframe.asp](http://www.w3schools.com/tags/tag_iframe.asp)

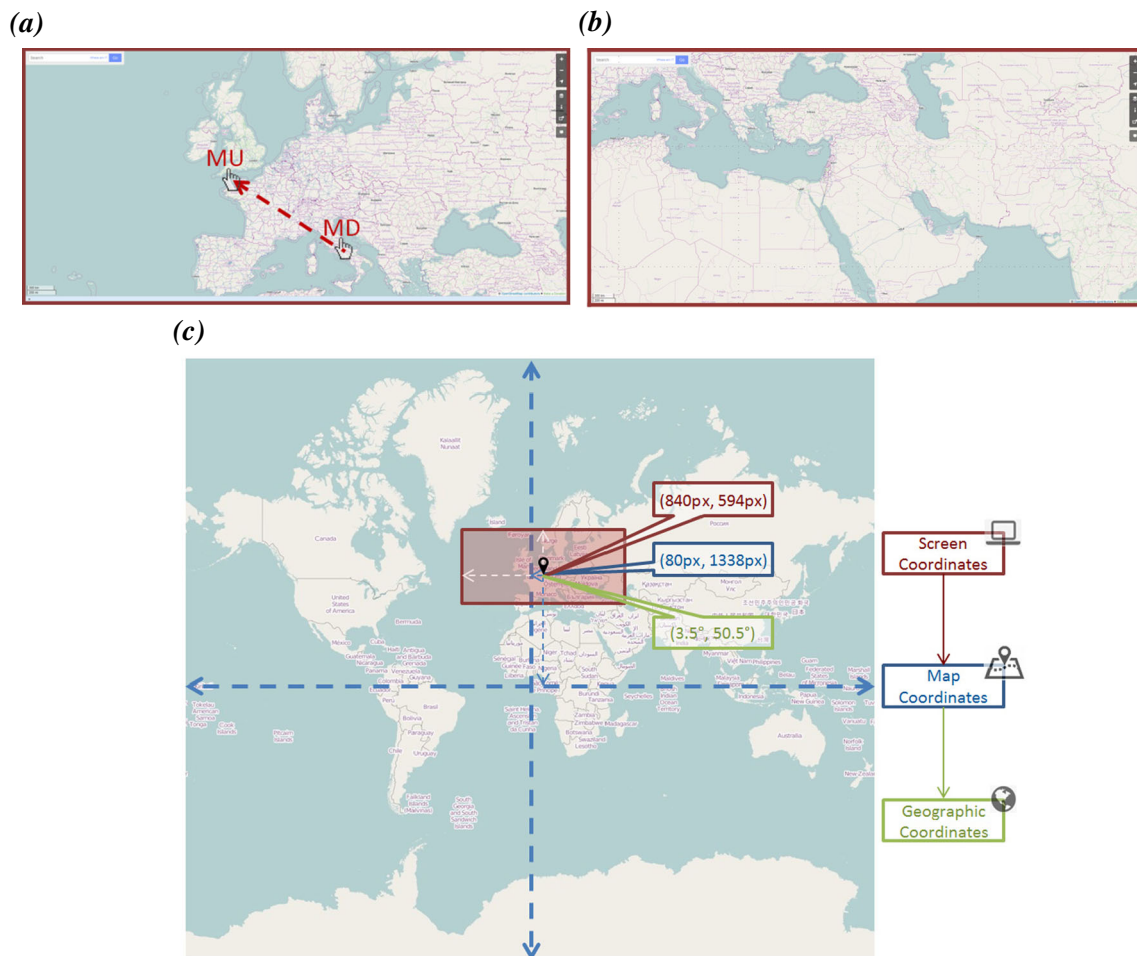
This problem can be solved using a proxy server—for example, PhProxy, as is explained in (Atterer, 2006; Atterer et al., 2006; Rodden & Fu, 2007). The page with the web map application is requested through the proxy, and as such, the parent and child pages can have the same domain. In this way, it is possible to log all user actions within an iFrame using JavaScript code attached to a parent page. Similarly, a proxy server can be used to directly attach a script to log user actions to a webpage, such as a web-mapping site, also without an iFrame. Since this solution is open-source, it is not linked to a specific eye tracking or web mapping application, and therefore, it will be implemented and further discussed herein. However, it must be recalled that this solutions works only with online (browser-based) applications.

We consider two different alternatives for the open source options—JNativeHook and PyHook—related, respectively, to the programming languages JAVA and Python. Both libraries request that the associated programming language be installed on the computer, with the necessary extensions able to ‘hook into’ the operating system. The combination of these building blocks—libraries, programming languages, extensions, and so forth—form our *desktop user-logging tool*. On the basis of the logged user actions, the corresponding geographic coordinates for each registered eye movement can be calculated (see next section). The code and manual for these libraries can be found on the following webpages: <http://sourceforge.net/apps/mediawiki/pyhook> and <http://code.google.com/p/jnativehook/>.

On the basis of the above considerations, we implemented and evaluated user-logging tools with an SDK, a proxy-server and the two desktop-based libraries. One of the desktop logging tools, which applies to the widest array of studies, is further evaluated in a number of case studies. These case studies allow us to observe how well the proposed methodology can be applied across multiple studies, including the proposed automatic geo-referencing of the obtained gaze coordinates for digital maps. The next section details how to transform the screen coordinates to geographic coordinates in the cases of a panning and a zooming operation. These two interaction types are considered because they are most often used (Harrower & Sheesley, 2005; Wilkening & Fabrikant, 2013)

#### Calculating geographic coordinates

The two main categories of user interactions that are possible on nearly any digital map are panning and zooming, each of which triggers a different response in the displayed image. On the basis of a detailed registration of the users’ interactions (time, distance, direction, and location), the recorded eye movements can be transformed to their associated geographic coordinates—that is, the eye movements can be georeferenced.



**Fig. 1** Illustrations of the panning operation and the associated coordinate systems. (a) Mouse-down (MD) and mouse-up (MU) locations for panning. (b) New position of the viewing window after the

panning operation. (c) Whole-map image showing the captured coordinates in three different coordinate systems

*Changing the extent of the map: Panning* The panning operation corresponds to moving a viewing window over the whole map image without changing the scale of the map. This *viewing window*, which corresponds to the screen on which the stimulus is presented, has its own reference system that consists of screen coordinates, typically expressed in pixels and relative to the upper left corner of the screen. To define the complete interaction for the panning operation, only the screen coordinates and the timestamps of the *mouse-key-down* (when the left mouse key is pressed) and *mouse-key-up* (when the left mouse key is released) events need to be registered. Within the time window between mouse key down and mouse key up (MD and MU in Fig. 1a), the map image is shifted in a certain direction. This is illustrated in Fig. 1a and b for the OpenStreetMap<sup>9</sup> online mapping application.

Because the scale of the map remains constant during a pan operation, it is possible to define every point on the map by a set of map coordinates (expressed in pixels) relative to the

center of the whole map image (Fig. 1). The screen coordinates of the center of the first viewing window, in pixels, are (840, 594), which are expressed relative to the red rectangle, thus illustrating the position and dimensions of the screen on which the map is presented. The corresponding map coordinates (in blue, relative to the center of the whole map) are (80, 1338), in pixels. Relative to the center of the viewing window, the map coordinates of all other pixels in the current window can be calculated.

In contrast to the screen and map coordinates, the geographic coordinates are related to a sphere that approximates the actual shape of the Earth. Consequently, to calculate the corresponding geographic coordinates, the associated map projection formulas are necessary, since they define the transformation of the coordinates from a sphere to a flat surface. Most popular mapping platforms, such as OpenStreetMap, Google Maps, Bing Maps, and MapQuest, use the *spherical Mercator* projection. The forward and inverse map projection formulas for the spherical Mercator projection are given in Table 1 (Snyder, 1987). The  $x$ - and  $y$ -coordinates in these

<sup>9</sup> [www.openstreetmap.org](http://www.openstreetmap.org)

**Table 1** Map projection formulas (forward and inverse) for the spherical Mercator projection

Forward Map Projection Formulas	Inverse Map Projection Formulas
$x = R(\lambda - \lambda_0)$	$\lambda = \lambda_0 + \frac{x}{R}$
$y = R \ln \left[ \tan \left( \frac{\pi}{4} + \frac{\varphi}{2} \right) \right]$	$\varphi = 2 \tan^{-1} \left[ \exp \left( \frac{y}{R} \right) \right] - \frac{\pi}{2}$

formulas correspond to the calculated map coordinates (in pixels). The value of  $R$  corresponds to the radius of a reduced sphere on which the projection is executed. Consequently,  $R$  reflects the scale of the map. To illustrate this, the calculated geographic coordinates of the center of the red rectangle in Fig. 1 are (3.5, 50.5) in *degrees*, corresponding to the geographic longitude ( $\lambda$ ) and latitude ( $\varphi$ ) of that position.

The screen coordinates of the illustrated panning operation are  $x = 1612$  and  $y = 954$  pixels for the mouse-down event, and  $x = 923$  and  $y = 418$  for the mouse-up event. This can be used to recalculate the map coordinates of the center of the new viewing window, at (769, 802) pixels at that particular scale level. On the basis of these new map coordinates of the center of the viewing window, the corresponding geographic coordinates can be calculated using the spherical Mercator formulas above.

*Changing the scale: Zooming* When zooming on a map image, the user changes the scale of the map. Online mapping applications typically work with a fixed set of predefined scale levels—that is, the multiscale map images are already rendered and stored in tiles. Every scale level represents the world at a certain size and level of detail. OpenStreetMap, for example, works with 20 levels of detail, each associated with a different value for  $R$  in the projection formulas.

Although other options (such as clicking + or – buttons, drawing a rectangle, etc.) are also available, the zooming operation is often controlled by the scrolling of the mouse wheel. When zooming in or out, the geographic and screen coordinates of the mouse position remain fixed. Because the scale level has changed, the associated value for  $R$  needs to be determined. This can be derived from *scroll wheel logging* (i.e., direction and number of ticks). Furthermore, the map coordinates of the new viewing window center must be calculated at the new scale level.

The geographic coordinates and screen coordinates of the mouse position during the scroll operation are both known and fixed and can, accordingly, be used for calculating the associated map coordinates at the new scale level (new  $R$ ) by using the forward map projection formula at that point. On the basis of the difference in screen coordinates between the mouse position and the center of the screen, the map coordinates of the viewing window center can be calculated. This makes it possible to calculate the corresponding geographic coordinates for all screen coordinates in the current viewing window

and that scale level. The next section will present an overview of potential logging tool that are implemented.

### Implementations of user logging and gaze georeferencing

An SDK-based solution for spatially referencing the gaze coordinates

As a vendor-specific example, Kuhn and Coltekin (2014) have implemented a solution to perform georeferenced gaze tracking based on Tobii's SDK. The main georeferencing was implemented as a C++ plugin of QuantumGIS.<sup>10</sup> The implementation is built according to a two-level approach, so it can work with multiple views using different projections synchronously. The first level (core part) delegates the incoming gaze data to various second-level modules. The delegation (or redirection) to the modules is based on the extent of the view (the rectangle), which every module needs to report. The program further contains a callback function that is activated when the core detects gaze data that intersect with the rectangle. This is how the program “knows” that the gaze is on this view. Once the gaze information is mapped to the currently visible screen extent, the gaze coordinates are georeferenced in real time and stored. Gaze data can then be supported with further information if and when needed. All collected data are then logged, along with the gaze data received from the eye tracker. With this modular system, it is possible to track a user's gaze while working with multiple independent or linked views side by side. The implementation was tested with 2-D and 3-D views, and in its particular form, it has various limitations, especially for 3-D viewing, where tilting complicates the interaction. In terms of computational performance, translating from screen coordinates to geographic coordinates is rather straightforward, and thus runs smoothly in real time on a modern computer. Certain complex typical geographical information system (GIS) functions, such as calculating an intersection between features, may take too long for real-time implementation, depending on the data source and the available indexes. However, for a 2-D map with a known setup (static set of shown layers and static symbology), it would be possible to perform such tasks in postprocessing in order to avoid delay. Another challenge is the accuracy of the eye tracker, which is not at a pixel level; therefore, it is not always possible to assign the current gaze to a single feature. Kuhn and Coltekin (2014), accordingly, recommended recording uncertainty parameters based on the zoom factor, along with the eye tracker data. Thus far, though, this implementation has its limitations in 3-D viewing, but it functions well in 2-D with zooming and panning when using Tobii SDK and associated applications.

<sup>10</sup> [www.qgis.org/](http://www.qgis.org/)

## Online logging through a proxy server

When implementing and testing the logging options using a proxy server, we discovered that certain online mapping sites (e.g., Bing Maps, MapQuest, etc.) block the registration of the mouse-down event on the map itself. The mouse-up event is registered on the map, whereas the mouse-down event is registered within the iFrame, but outside the map image. Consequently, not all online mapping sites can be tested with this setting. In addition, we found a similar problem when using a proxy server that directly loads the mapping site without iFrames and attaches JavaScript code (for user logging) to it. Hence, the registration of the mouse-down event is blocked, but only on the map image. Because of its limited usefulness, this approach is not considered further. In the following sections, the desktop-based user-logging tools will be evaluated.

## Desktop logging with open-source libraries

The JNativeHook and PyHook libraries were tested in a user study (Dupont, Pihel, Ode, & Van Eetvelde, 2013; see Case Study 1 for further details on the test). The original code of these libraries (JNativeHook and PyHook) was adapted to the experimenter's needs (i.e., to facilitate the analyses afterward): additional information was requested from the experimenter (e.g., a participant's ID), the registered data were written in a specific structure (column headings) and format (csv), and the data were saved in an appropriate folder. When using JNativeHook, we found that not all user events were registered correctly. More specifically, pressing the spacebar to go to the next stimulus was not always recorded by the logging tool, a problem that considerably complicated the analyses. This failure to record could be the result of a conflict with the eye tracking software that was recording the participants' eye movements at the same time, at a rate of 120 Hz. However, this issue was not encountered when executing the same test using PyHook, since all data were properly recorded with this system.

## Logging tool selection based on applicability

In summary, among our implementation experiments, the tool that could be applied to the widest array of applications was the *desktop-based user-logging tool* with the *PyHook library*. PyHook allows for the logging of user actions on webpages, independent of the API and the source of the information, and on desktop applications. The main downside of this method is synchronization with the eye tracking device. Since the library is not linked with the eye tracker itself, the timestamps in the recordings do not correspond. Therefore, a synchronization point (e.g., an imposed mouse click that is registered by both systems) must be predetermined before initiating the recordings with both tools. A number of case studies are presented below in which we test the combination of eye tracking and

user logging. The experiments were repeated using eye tracking devices from three important vendors—SMI, Tobii, and SR Research<sup>11</sup>—in order to check its applicability with respect to these different devices. Furthermore, georeferencing eye movement data opens up new possibilities for data analyses. Next, the combined user-logging and eye tracking methodology is applied in user research with static stimuli (maps and photographs). Finally, the applicability of the proposed logging methodology in other fields will be considered in the Discussion section.

## Case studies

### Case Study 1: Evaluating JNativeHook and PyHook

In this case study, both desktop-based user-logging tools (related to JNativeHook and PyHook) were evaluated. In a first step, only static stimuli (photographs) were included in a user study, to be able to verify the suitability and accuracy of both tools. In the next study (see Case Study 2), interactive (cartographic) stimuli were included to evaluate the georeferencing methodology.

During the initial case study, the participants' eye movements were recorded while they were looking at photographs of different landscapes. In total, 63 landscapes were presented to the participants. For each image, the participants were asked to indicate the region in the photograph they found most eye-catching by drawing a rectangle over that region. After completing this task, the spacebar was pressed to continue to a questionnaire in which the participants were asked to indicate why they found that part of the image eye-catching. After the participant had pressed OK, the next picture was presented. Before the start of the actual test and after the calibration, the participants were asked to press a button on the screen. This action synchronized the timestamps from the eye tracking device and the user-logging tool. The study itself is described in more detail in a previous article by Dupont et al. (2013b).

The test was conducted with the SMI RED eye tracker in the Eye Tracking Laboratory of the Department of Geography at Ghent University. During the initial main test, JNativeHook was used to log the participants' mouse and keyboard actions. Yet, data analysis revealed that not all spacebar actions were recorded by this logging tool, which significantly complicated the analyses. The test was executed again, but this time the PyHook library was used to log user actions. In this case, all of the data were recorded properly. Figure 2 shows an extract from the logging dataset in which a clear pattern is visible: first drawing a rectangle (mouse down and up; rows in shades of orange), then pressing the space bar (in black), indicating an answer in the questionnaire, and pressing OK (in shades of

<sup>11</sup> [www.sr-research.com/](http://www.sr-research.com/)



Type	MsgID	MsgName	Message	Time	Injected	PosX	PosY	Wheel	Ascii	Key	KeyID	ScanCode	Ext	Alt	Transit	Wndw	WndwName	WndwRect	WndwClassName
MOUSE	1355	mouse left down	513	2156963	0	1877	394	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	1500	mouse left up	514	2158335	0	2606	617	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
KEY	1532	key down	256	2160177	0	None	None	None	32	Space	32	57	0	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	1871	mouse left down	513	2164966	0	1968	332	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	1872	mouse left up	514	2165075	0	1968	332	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	1949	mouse left down	513	2166627	0	1995	662	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	1950	mouse left up	514	2166136	0	1995	662	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	2195	mouse left down	513	2170332	0	2270	320	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	2382	mouse left up	514	2172501	0	2742	497	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
KEY	2383	key down	256	2173468	0	None	None	None	32	Space	32	57	0	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	2569	mouse left down	513	2175870	0	1965	278	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	2570	mouse left up	514	2175979	0	1965	278	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	2633	mouse left down	513	2176744	0	2021	669	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	2634	mouse left up	514	2176837	0	2021	668	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	2885	mouse left down	513	2181751	0	2050	541	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	3039	mouse left up	514	2183826	0	2495	682	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
KEY	3040	key down	256	2185418	0	None	None	None	32	Space	32	57	0	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	3141	mouse left down	513	2187056	0	1995	272	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	3142	mouse left up	514	2187165	0	1995	272	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	3224	mouse left down	513	2188163	0	1997	664	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	3225	mouse left up	514	2188272	0	1997	664	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	3412	mouse left down	513	2192157	0	2223	844	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	3515	mouse left up	514	2193274	0	2501	971	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
KEY	3519	key down	256	2194029	0	None	None	None	32	Space	32	57	0	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	3853	mouse left down	513	2198210	0	1968	307	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	3855	mouse left up	514	2198288	0	1967	307	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	3918	mouse left down	513	2199036	0	2013	667	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	3919	mouse left up	514	2199146	0	2013	667	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	4102	mouse left down	513	2202858	0	2000	612	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
MOUSE	4173	mouse left up	514	2204969	0	2084	978	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1
KEY	4198	key down	256	2205105	0	None	None	None	32	Space	32	57	0	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	4553	mouse left down	513	2208693	0	1959	260	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	4559	mouse left up	514	2208802	0	1959	262	0	None	None	None	None	None	None	None	132134	SMI_Experiment	(1600, 0, 3280, 1050)	HwndWrapper[SMI Experiment Center.exe;cbf443d7
MOUSE	4573	mouse left down	513	2209223	0	1960	279	0	None	None	None	None	None	None	None	852992	None	(1600, 0, 3280, 1050)	WindowsForms10.Window.8.app.0.f1357c..._113_ad1

Fig. 2 Filtered output of the user-logging tool (based on PyHook); rows are ordered by time. The color-coded rows indicate registered user actions

green). The light colors correspond to the left mouse-key-down actions, and the darker colors correspond to the left mouse-key-up actions.

The user-logging data reveal that when the participant begins drawing the rectangle, this action results in a quantitative measurement that indicates how long the participants will need to make a decision. In addition, the rectangle (position and size) is recorded through the screen coordinates of the mouse-down and mouse-up actions. We wrote a script that can

read a list of subsequent mouse-down and mouse-up actions in CSV format and translate the data into an XML file that can be imported into BeGaze, SMI’s software to analyze eye tracking data. As such, the user-generated rectangles can be used as AOIs on which further analyses can be conducted—for example, determining numbers of fixations, dwell times inside/outside the AOI, and overlap between the AOIs. A resulting AOI with statistics in BeGaze and its associated XML file are depicted in Fig. 3.

```
<?xml version='1.0'?>
<ArrayOfDynamicAOI xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
<DynamicAOI>
<ID>1</ID>
<ParentID>1</ParentID>
<TrackingUsage>None</TrackingUsage>
<Group />
<Enabled>true</Enabled>
<Scope>Local</Scope>
<Transparency>50</Transparency>
<Points>
<Point>
<X>321</X>
<Y>850</Y>
</Point>
<Point>
<X>979</X>
<Y>397</Y>
</Point>
</Points>
<BorderWidth>2</BorderWidth>
<Type>Rectangle</Type>
<Style>HalfTransparent</Style>
<HatchStyle>DarkDownwardDiagonal</HatchStyle>
<Color>NamedColor:DarkSlateBlue</Color>
<Name>test_5</Name>
<Font>
<FontName>Tahoma</FontName>
<FontSize>13</FontSize>
<FontStyle>Regular</FontStyle>
<FontUnit>Point</FontUnit>
<FontGdiCharSet>1</FontGdiCharSet>
<FontGdiVerticalFont>false</FontGdiVerticalFont>
</Font>
<ReferenceStimulusID>0</ReferenceStimulusID>
<MovieFrameRate>0</MovieFrameRate>
<PlaneGridRows>0</PlaneGridRows>
<PlaneGridColumns>0</PlaneGridColumns>
<Visible>true</Visible>
<CurrentTimestamp>0</CurrentTimestamp>
<KeyFrames />
<TrackedKeyFrames />
</DynamicAOI>
</ArrayOfDynamicAOI>
```

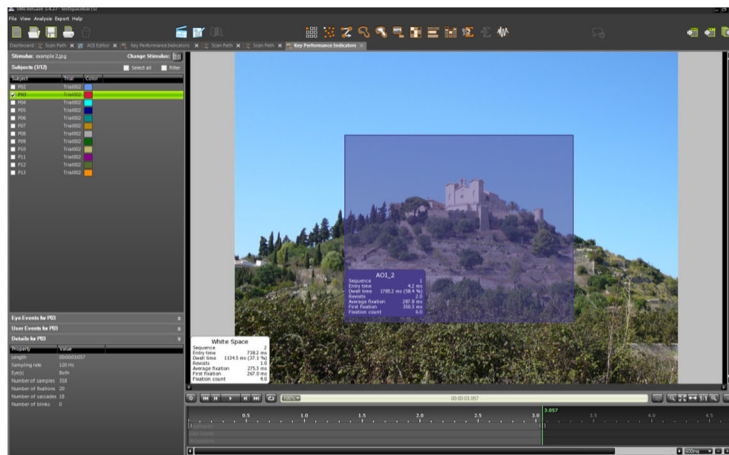


Fig. 3 Conversion of a rectangle to an XML file and visualization of the associated area of interest in SMI BeGaze

**Table 2** Characteristics of the three eye tracking devices used in Case Study 2

Vendor	SMI	Tobii	SR Research
Name of eye tracker	RED250	T120	EyeLink1000
Type	Remote	Remote	Desktop mounted with chin rest
Location	Department of Geography, Ghent University	Department of Geography, University of Zurich	Department of Experimental Psychology, Ghent University
Sampling rate	60–120 Hz	60–120 Hz	1000 Hz
Monitor	22 inch (1680 × 1050 px)	22 inch (1920 × 1080 px)	21 inch (1024 × 768 px)

### Case Study 2: Georeferencing with PyHook

In the next case study, the online OpenStreetMap (OSM) mapping platform was used as a test platform in a pilot experiment. The pilot experiment was repeated using three different eye tracking devices, whose main characteristics are presented in Table 2. As a consequence, three trials with eye movement and user-logging data are recorded, each of which is associated with different hardware (eye tracking devices). The monitors attached to the three different eye tracking systems all had different resolutions, but this did not influence the experiment itself, but rather only had to be taken into account during the

subsequent georeferencing process. Web mapping sites, such as the OSM, typically work with a number of *zoom levels* or *scale levels*. When panning, the scale level remains the same. When zooming in or out, however, the scale level changes. The OSM has 20 fixed scale levels, numbered 0 to 19.

After the calibration process, participants were asked to push a button that allowed for synchronization of the time measurements from the eye tracker and the PyHook-based logging tool. The mouse-down action in the logging tool corresponded to a mouse click action in the eye tracking software. The screen recording mode was then activated (or the pop-up calibration mode, for the EyeLink 1000) and a

**Fig. 4** Participant task: Initial map image and subsequent panning locations

URL ([www.openstreetmap.org/#map=5/50.000/3.500](http://www.openstreetmap.org/#map=5/50.000/3.500)) was loaded into the browser. This URL contains the scale level (5, which corresponds to a scale of 1:15,000,000) and the geographic coordinates of the center point of the current viewing window in OpenStreetMap (50 deg north and 3.5 deg east). This corresponds to the first image in Fig. 4 (which might have differed slightly due to screen resolutions). The participant was then asked to pan to the different regions depicted in the assignment (see Fig. 4).

The eye movements recorded during the three trials (on the three different eye trackers) were exported using each of the associated software packages and aggregated into fixations. Figure 5 shows the fixations of one participant who was tested with the eye tracking device from SR Research. This image presents the locations on the screen where the participant was fixating, but it does not reveal where on the map the fixations took place. The screen coordinates that locate the fixations are then transformed to map coordinates, and finally to geographic coordinates, according to the descriptions in the previous sections.

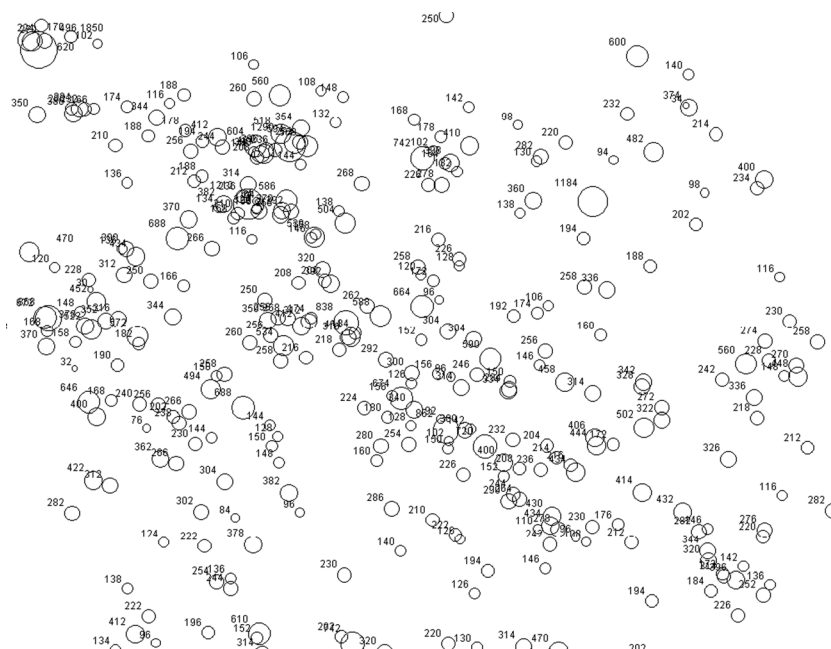
The resulting georeferenced fixations could then be imported into GIS software (i.e., ArcGIS) and placed on top of a dataset that depicts the world's continents (see Figs. 6 and 7). Hence, all tools and functions available in a GIS could be applied to the imported fixations and used to analyze these fixations. The picture depicted in Fig. 6 shows, for example, all imported fixations reprojected in the spherical Plate Carrée map projection. In Fig. 7, the spherical Mercator projection, which is also used in OSM, is applied (Snyder, 1987). Reprojecting the data might yield useful insights into how eye movements are influenced by distortions in the map image

due to different projection systems. Figure 7a shows a *buffer* operation (a typical GIS operation), whereby a polygon is drawn around the fixations (only from the SMI eye tracker, in this case). All points in each polygon are within 500 km of the fixation points. This operation was repeated for the fixations of the three trials. With the *intersect* operation (see Fig. 7b), the overlapping zones among the three polygons were calculated and added as a new layer to the dataset. The resulting polygons (in beige) correspond to the world regions to which the participants were instructed to pan (see Fig. 4).

When studying the time measurements from the eye tracker and the logging tool, we discovered that small deviations between the two exist. Since these were not unidirectional, they were associated with the actual time registration mechanisms for both. However, a maximal deviation of 10 ms was registered, which was acceptable when taking into account the sampling rates of the eye tracking devices from SMI and Tobii (120 Hz, or every 8.33 ms, for both).

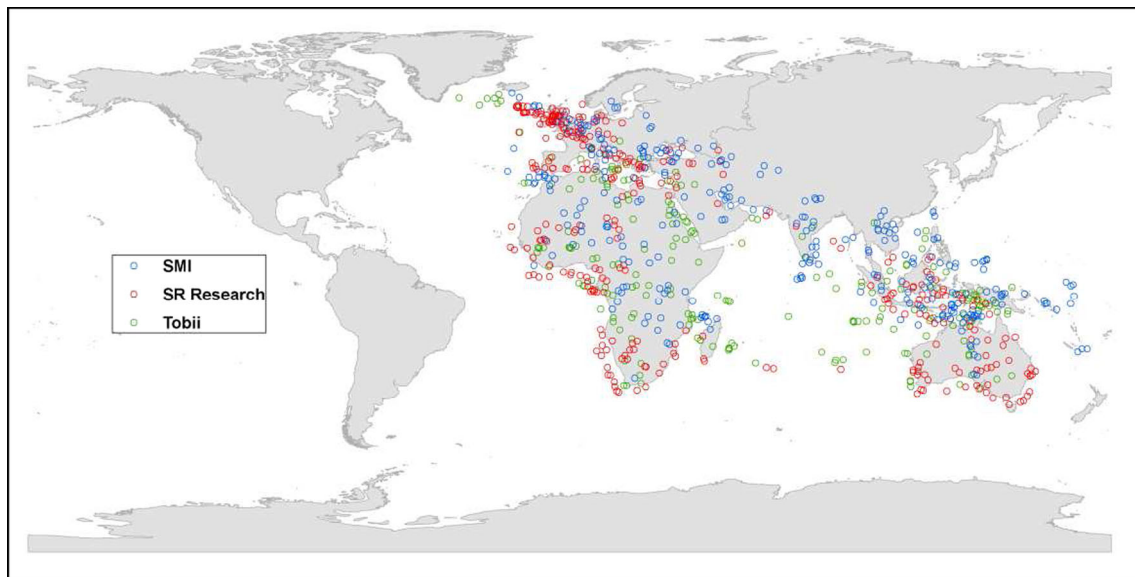
## Discussion

The pilot experiments suggest that the methodology that we proposed can be used consistently across various eye tracker hardware and software setups to transform recorded gaze coordinates, expressed in screen coordinates, automatically into geographic coordinates. Our suggested approach is based on freely available and open-source software, and therefore can be used independently of the type of eye tracker, as well as with static stimuli (see Case Studies 1 and 2). The



**Fig. 5** Sample fixation plot of one participant recorded with the SR Research eye tracker, based on screen coordinates



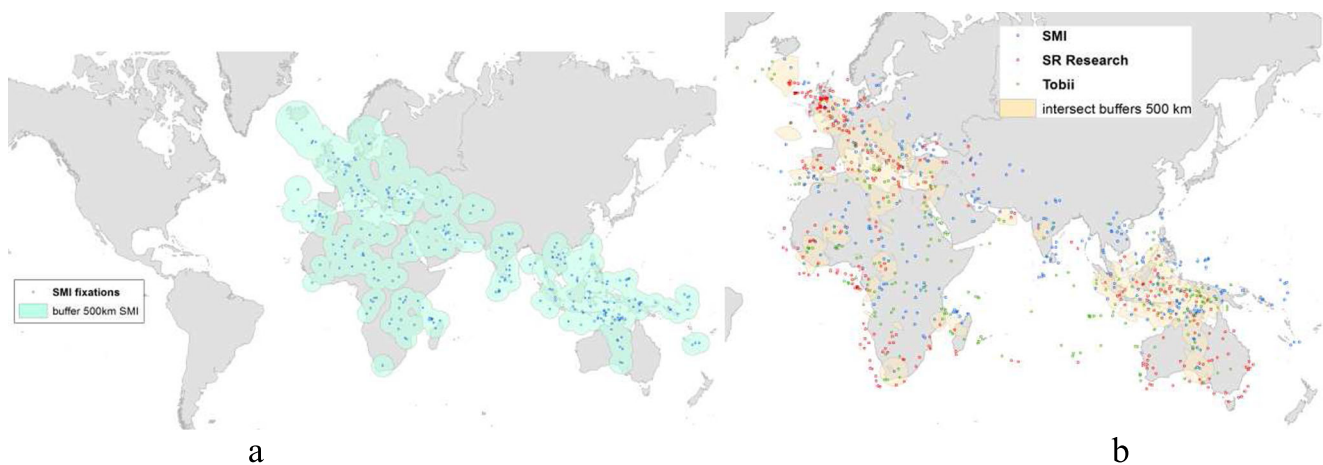


**Fig. 6** Fixation data from three eye trackers, imported into a geographical information system (GIS) and displayed with the spherical plate Carrée map projection

synchronization issue between the time recordings of the logging tool and the eye tracker is a minor issue and could be overcome, for example, by including a keypress action at the start of the study that is recorded by both the eye tracker and the logging tool. Accordingly, this keypress would serve as a reference point in time. Our approach and the associated validation studies (the presented case studies) offer efficient analyses and comparisons for other user studies through which static and dynamic stimuli could be evaluated.

The recordings from the user-logging tool and subsequent conversion of the screen coordinates to geographic coordinates are a vital aid to be able to analyze data without much manual interference. The obtained data can be automatically queried on the basis of a number of criteria, which are explained below:

- Query the eye tracking data on the basis of *screen coordinates*: Through this analysis, it can be determined where on the screen the users focus their attention (e.g., more on one side of the screen than the other). This, however, is a standard practice for which the additional logging tool is not required. However, for interactive applications, this level of querying might not suffice.
- Query the eye tracking data on the basis of *map coordinates*: This takes the distortions introduced by the map projections into account. These coordinates (in pixels), however, are dependent on the scale level. Although this works for panning operations, when one is working across multiple zoom levels, additional calculations are necessary. This practice is not novel, though it is applied only rarely. Nevertheless, the proposed logging



**Fig. 7** Fixation data from three eye trackers, imported into geographical information system (GIS), displayed with the spherical Mercator projection and (a) a buffer (500 km) around each point and (b) an intersect operation



tool could facilitate automatic calculations, and thus increase its usability.

- Query the eye tracking data on the basis of *geographic coordinates*: The calculated geographic coordinates are independent of the scale level. As such, how users visualize or perceive the different parts of the world (or rather, their associated visualization) can be explored. This has rarely if ever been done, because it is technically complex and is facilitated by the proposed approach.
- Query the eye tracking data on the basis of scale level: This is associated with the zooming action. Consequently, users' attentive behaviors and how the behaviors vary across a number of scale levels can be evaluated. This query can be executed without georeferencing the eye movement data; that is, only the scale level has to be determined.
- Query the eye tracking data on the basis of other interactions: This can be used to compare the eye movement metrics before, during, and after an interaction (e.g., to determine how participants process the information). As we previously discussed, certain eye tracking software packages from commercial vendors make it possible to register these interaction, though often not on a detailed level (e.g., to distinguish between mouse-up and mouse-down events). However, these solutions are vendor-specific, which limits their applicability in comparison to the proposed solution.

Furthermore, the queried or filtered data can be imported into a GIS based on the geographic coordinates, which means that they can be included in analyses that are typically available in a GIS package, such as buffer and cluster analyses, among others (similarly to, e.g., Coltekin et al., 2009; Li, Coltekin, & Kraak, 2010).

Georeferenced gaze coordinates offer incremental yet important progress in the current analysis of eye movement and the analysis of interactive and dynamic map stimuli. Using current analysis routines, most often the interactions must be located manually on the basis of a video recording of the screen. Next, the data must be (manually) segmented and labeled (i.e., by scale level or geographic region). This allows for an analysis and comparison of the data, but the exact geographical positions of the gaze coordinates (points of regard) remain unknown. The geographic coordinates facilitate the comparison of the participants' eye movements on a certain geographic location, visited at a different timestamp, because of the interaction tools. Furthermore, the geographic coordinates can be imported into a GIS, in which a wide array of functions are available for further analyses. Similar issues arise when using existing solutions such as dynamic AOIs or semantic gaze mapping. Analyses are often executed either on a *qualitative* level, at which eye movements are described for each participant separately and compared as such, or on an *analytical* level, whereby the analyst must engage in large

amounts of laborious manual work, which could be avoided through an automated process such as the one proposed herein. In the next paragraphs, the applicability of the selected user-logging method (based on PyHook) is discussed.

#### Applications in other research fields

In most cases, possible interactions in a user study can be classified into two types of behavior, as illustrated by the case studies in this article:

1. The participant can interact freely with a given system: clicking, dragging, zooming, and so forth. These actions cause some reaction (e.g., open a menu, zoom in on a region, go to a new webpage) that can be reconstructed and queried when logging the mouse and keyboard actions.
2. The participants may be asked to indicate a region of interest. This can be more complex than clicking, such as when drawing a rectangle around an area. This shape (e.g., the rectangle) can be translated into an AOI, which can then be analyzed or be used in an analysis of eye movements.

The methodological problems that arise from these interactions also occur in other research fields in which eye tracking is used with interactive applications, such as experimental psychology (e.g., Allopenna et al., 1998; Rayner, 2009; Reichle, Warren, & McConnell, 2009; Van Assche, Drieghe, Duyck, Welvaert, & Hartsuiker, 2011; Van der Haegen, Cai, Stevens, & Brysbaert, 2013), marketing research (Pieters, 2008; Pieters & Wedel, 2004; Wedel & Pieters, 2006; e.g., Chandon, Hutchinson, Bradlow, & Young, 2009; Cian, Krishna, & Elder, 2013; Townsend & Kahn, 2014), sports and movement sciences (e.g., Lenoir et al., 2000; Vaeyens, Lenoir, Williams, & Philippaerts, 2007; Vansteenkiste, Cardon, D'Hondt, Philippaerts, & Lenoir, 2013; Vansteenkiste, Vaeyens, Zeuwts, Philippaerts, & Lenoir, 2014), and so forth. Therefore, the user-logging methodology that is proposed in this article can also be beneficial in these research fields, especially when spatial referencing is possible—which, in most cases, it is.

In the field of traffic science, for example, the logging tool could be used to analyze data from hazard perception tests. In a hazard perception test (Crundall, Chapman, Phelps, & Underwood, 2003; Vansteenkiste, Zeuwts, Cardon, & Lenoir, 2013), participants usually have to click with the mouse on potential hazards in videos of traffic situations. When this paradigm is combined with eye tracking, the logging tool would allow a detailed registration of the user actions, making it possible to link them to the eye tracking data. Similarly, the tool could be useful in linking eye movements and steering behaviors while driving in a simulator. In sports sciences, the logging tool, in combination with

eye tracking, could be used in a tactical decision-making task (Vansteenkiste et al., 2013b) or in an error analysis task. For example, when judging a video of a gymnastic performance, the actions of a judge (such as replaying a video, zooming and panning on an image, indicating zones of interest in an image, etc.) can all be registered.

A similar application also exists in marketing. For example, in digital promotion folders, it is possible to flip to the next page, zoom in on a specific item on a page, pan across the detail image, zoom out again, and so forth. Although very few research reports are available regarding these new marketing tools, such research could be facilitated using the methodology described in this article. Online maps (e.g., Google, Bing) can, in this context, also be used as a marketing tool, since millions of consumers around the world already use Google Maps when searching for the perfect restaurant, checking out the best hotels, or finding the nearest ATM. Google Maps is becoming a virtual marketplace, since business owners can easily list themselves on Google Maps and display useful information about their businesses and their services.

With respect to work and organizational psychology, the logging tool can be applied in a number of test cases. For instance, researchers at the career-matching company TheLadders tested how a CV or letter of application is scanned to search for information on a certain candidate (e.g., on the Web; TheLadders, 2012). This could be extended with the logging tool, enabling participants to indicate what they find to be of interest to them. Finally, the proposed tool could be used to collect eye movement data while the subjects' cognitive performance is being assessed. For example, eye movement research on text comprehension could be enriched with information as students highlight important keywords or sentences in a textbook passage (e.g., Ponce, López, & Mayer, 2012). In the visuospatial abilities domain, the methodology could facilitate investigating ocular information while subjects navigate through a virtual maze environment (e.g., Akinlofa, Holt, & Elyan, 2014).

### Conclusion and future work

This article describes a methodological framework that can be used to efficiently and systematically evaluate interactive applications that can be spatially referenced to real world coordinate systems (such as in cartography) by applying a combination of eye tracking and user logging. By logging the users' actions in detail, the output of the eye tracker—gaze position or point of regard, expressed in screen coordinates—can be transformed to geographic coordinates. This facilitates analyses of the data, which, as such, can then be largely automated. By allowing researchers to bypass tedious, and often manual, selection and structuring of the data, a more

automatized approach of the analyses is made possible. In addition, this methodology appears to be potentially beneficial for a number of other research fields.

Nevertheless, the methodology should be further optimized. First, it is still difficult to derive the current scale level after a zooming action, since the number of scroll “clicks” does not correspond to the number of scale levels traversed. Thus, further research will be needed to determine the number of scroll “clicks” that cause the scale level to change, in relation to the speed with which this action is performed. Second, it should also be possible to evaluate other digital cartographic applications, such as Google Earth. This means that the correct projection formulas must be determined, taking into account that the projection can also be rotated in all directions by the user. This can be extended to other applications that may not visualize a part of the Earth. An interesting example is an evaluation of the usability of the space–time cube, as described by Kveladze, Kraak, and van Elzakker (2013). The space–time cube is often visualized as a 3-D interactive system with which users can interact. That is, the users can rotate, zoom in or out, change the layers that are visualized, change the time filters, and so forth. Logging these user actions in combination with the participants' eye movements would yield critical information with regard to the use of this application.

As a follow-up to this project, the proposed methodological framework will be implemented in user studies whose goal will be to evaluate interactive digital cartographic products, such as online mapping sites (e.g., OSM, Google Maps, or more complex mashup maps), and to assess the impacts of the interactive tools on the (different types of) map users' cognitive processes.

### References

- Akinlofa, O. R., Holt, P. O.'B., & Elyan, E. (2014). The cognitive benefits of dynamic representations in the acquisition of spatial navigation skills. *Computers in Human Behavior*, *30*, 238–248. doi:10.1016/j.chb.2013.09.009
- Allopenna, P. D., Magnuson, J. S., & Tanenhaus, M. K. (1998). Tracking the time course of spoken word recognition using eye movements: Evidence for continuous mapping models. *Journal of Memory and Language*, *38*, 419–439.
- Atterer, R. (2006, May). *Logging usage of AJAX applications with the “UsaProxy” HTTP proxy*. Article presented at the WWW2006 Workshop on Logging Traces of Web Activity: The Mechanics of Data Collection, Edinburgh, Scotland.
- Atterer, R., Wnuk, M., & Schmidt, A. (2006, May). *Knowing the user's every move: User activity tracking for website usability evaluation and implicit interaction*. Paper presented at the 15th International World Wide Web Conference, Edinburgh, Scotland.
- Brewster, S. (2002). Overcoming the lack of screen space on mobile computers. *Personal and Ubiquitous Computing*, *6*, 188–205.

- Brodersen, L., Andersen, J. H. K., & Weber, S. (2001). Applying the eye-movement tracking for the study of map perception and map design. In *Kort and Matrikelstyrelsen* (p. 98). Copenhagen, Denmark: National Survey & Cadastre.
- Buswell, G. T. (1935). *How people look at pictures* (pp. 142–144). Chicago, IL: University of Chicago Press.
- Cartwright, W. (2012). Neocartography: Opportunities, issues and prospects. *South African Journal of Geomatics*, 1, 14–31.
- Castner, H. W., & Eastman, J. R. (1984). Eye-Movement Parameters and Perceived Map Complexity .1. *American Cartographer*, 11(2), 107–117.
- Castner, H. W., & Eastman, J. R. (1985). Eye-Movement Parameters and Perceived Map Complexity .2. *American Cartographer*, 12(1), 29–40.
- Chandon, P., Hutchinson, J. W., Bradlow, E. T., & Young, S. H. (2009). Does in-store marketing work? Effects of the number and position of shelf facings on brand attention and evaluation at the point of purchase. *Journal of Marketing*, 73, 1–17.
- Cian, L., Krishna, A., & Elder, R. S. (2013). This logo moves me: Dynamic imagery from static images. *Journal of Marketing Research*, 51, 184–197.
- Coltekin, A., Demsar, U., Brychtova, A., & Vandrol, J. (2014, September). *Eye-hand coordination during visual search on geographic displays*. Paper presented at the 2nd International Workshop on Eye Tracking for Spatial Research, Vienna, Austria.
- Coltekin, A., Fabrikant, S. I., & Lacayo, M. (2010). Exploring the efficiency of users' visual analytics strategies based on sequence analysis of eye movement recordings. *International Journal of Geographical Information Science*, 24, 1559–1575.
- Coltekin, A., Heil, B., Garlandini, S., & Fabrikant, S. I. (2009). Evaluating the effectiveness of interactive map interface designs: a case study integrating usability metrics with eye-movement analysis. *Cartography and Geographic Information Science*, 36, 5–17.
- Crundall, D., Chapman, P., Phelps, N., & Underwood, G. (2003). Eye movements and hazard perception in police pursuit and emergency response driving. *Journal of Experimental Psychology: Applied*, 9, 163–174. doi:10.1037/1076-898X.9.3.163
- Das, T., van Elzakker, C. P. J. M., & Kraak, M.-J. (2012, September). *Conflicts in neogeography maps*. Paper presented at AutoCarto 2012, Columbus, Ohio.
- Dobson, M. W. (1977). Eye movement parameters and map reading. *Cartography and Geographic Information Science*, 4(1), 39–58.
- Dong, W., Liao, H., Roth, R. E., & Wang, S. (2014). Eye tracking to explore the potential of enhanced imagery basemaps in web mapping. *Cartographic Journal*, 51(4), 313–329 doi:10.1179/1743277413Y.0000000071
- Duchowski, A. T. (2007). *Eye tracking methodology—Theory and practice*. Berlin, Germany: Springer.
- Dupont, L., Antrop, M., & Van Eetvelde, V. (2013a). Eye-tracking analysis in landscape perception research: Influence of photograph properties and landscape characteristics. *Landscape Research*, 39, 417–432.
- Dupont, L., Pihel, J., Ode, A., & Van Eetvelde, V. (2013, September). *Analyzing the perception of water surfaces in urban landscapes using eye tracking*. Paper presented at the IALE 2013 European Congress, Manchester, UK.
- Fabrikant, S. I., & Lobben, A. (2009). Introduction: Cognitive issues in geographic information visualization. *Cartographica*, 44, 139–143.
- Fabrikant, S. I., Rebich-Hespanha, S., Andrienko, N., Andrienko, G., & Montello, D. R. (2008). Novel method to measure inference affordance in static small-multiple map displays representing dynamic processes. *Cartographic Journal*, 45, 201–215.
- Fits, P. M., Jones, R. E., & Milton, J. L. (1950). Eye movements of aircraft pilots during instrument-landing approaches. *Aeronautical Engineering Review*, 9, 24–29.
- Garlandini, S., & Fabrikant, S. I. (2009). Evaluating the effectiveness and efficiency of visual variables for geographic information visualization. In *Spatial information theory* (pp. 195–211). Berlin, Germany: Springer.
- Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N., & Wichansky, A. M. (2002, March). *Eye tracking in web search tasks: Design implications*. Paper presented at the Eye Tracking Research and Applications Symposium, New Orleans, LA.
- Haklay, M., Singleton, A., & Parker, C. (2008). Web Mapping 2.0: The neogeography of the GeoWeb. *Geography Compass*, 2, 2011–2039.
- Harrower, M. (2007). The cognitive limits of animated maps. *Cartographica*, 42, 349–357.
- Harrower, M., & Sheesley, B. (2005). Designing better map interfaces: A framework for panning and zooming. *Transactions in GIS*, 9, 77–89.
- Hazzard, E. (2011). *OpenLayers 2.10 beginner's guide*. Birmingham, UK: Packt.
- Hegarty, M., Canham, M. S., & Fabrikant, S. I. (2010). Thinking about the weather: How display salience and knowledge affect performance in a graphic inference task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 36, 37–53. doi:10.1037/a0017683
- Hilbert, D. M., & Redmiles, D. F. (2000). Extracting usability information from user interface events. *ACM Computing Surveys*, 32, 384–421.
- Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Van de Weijer, J. (2011). *Eye tracking: A comprehensive guide to methods and measures*. Oxford, UK: Oxford University Press.
- Incoul A, Ooms, K., De Maeyer P. (2015). Comparing paper and digital topographic maps using eye tracking. In: A. Vondrakova, J. Brus, & V. Vozenilek (Eds.) *Modern Trends in Cartography. Lecture Notes in Geoinformation and Cartography* (pp. 534). Berlin: Springer.
- Ivory, M. Y., & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, 33, 470–516.
- Jacob, R., & Karn, K. (2003). Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In R. Radach, J. Hyönä, & H. Deubel (Eds.), *The mind's eye: Cognitive and applied aspects of eye movement research* (pp. 573–605). Amsterdam, The Netherlands: Elsevier.
- Kiefer, P., Giannopoulos, I., & Raubal, M. (2014). Where am I? Investigating map matching during self-localization with mobile eye tracking in an urban environment. *Transactions in GIS*, 18(5), 633–791.
- Kraak, M. J., & Brown, A. (2001). *Web cartography: Developments and prospects*. New York, NY: Taylor & Francis.
- Kuhn, M., & Coltekin, A. (2014). *Geocoded gaze data* (Working paper). University of Zurich, Department of Geography.
- Kveladze, I., Kraak, M.-J., & van Elzakker, C. P. (2013). A methodological framework for researching the usability of the space-time cube. *Cartographic Journal*, 50, 201–210.
- Lenoir, M., Crevits, L., Goethals, M., Duyck, P., Wildenbeest, J., & Musch, E. (2000). Saccadic eye movements and finger reaction times of table tennis players of different levels. *Neuro-Ophthalmology*, 24, 335–338.
- Li, X., Coltekin, A., & Kraak, M. J. (2010). Visual exploration of eye movement data using the space-time-cube. In *Geographic information science* (pp. 295–309). Berlin, Germany: Springer.
- Luebbering, C. R., Carstensen, L. W., Campbell, J. B., & Grossman, L. S. (2008). Expanding display size and resolution for viewing geospatial data: A user study with multiple-monitor high-resolution displays. *Cartography and Geographic Information Science*, 35, 203–219.
- MacEachren, A. M. (1995). *How maps work: Representation, visualization, and design*. New York, NY: Guilford Press.



- MacEachren, A. M., & Kraak, M.-J. (2001). Research challenges in geovisualization. *Cartography and Geographic Information Science*, 28, 3–12.
- Montello, D. R. (2009). Cognitive research in GIScience: Recent achievements and future prospects. *Geography Compass*, 3, 1824–1840.
- Moseme, M. T., & van Elzakker, C. P. J. M. (2012, September). *Neogeography map users and use*. Paper presented at AutoCarto 2012, Columbus, Ohio.
- Nielsen, J. (1993). *Usability engineering*. San Francisco, CA: Morgan Kaufmann.
- Nivala, A.-M., Sarjakoski, L. T., Jakobsson, A., & Kaasinen, E. (2001, August). *Usability evaluation of topographic maps in mobile devices*. Paper presented at the 20th International Cartographic Conference, Beijing, China.
- Ooms, K., de Maeyer, P., Fack, V., Van Assche, E., & Witlox, F. (2012). Investigating the effectiveness of an efficient label placement method using eye movement data. *Cartographic Journal*, 49, 234–246.
- Paganelli, L., & Paternò, F. (2002, January). *Intelligent analysis of user interactions with web applications*. Paper presented at the 7th International Conference on Intelligent User Interfaces, San Francisco, California.
- Papenmeier, F., & Huff, M. (2010). DynAOI: A tool for matching eye-movement data with dynamic areas of interest in animations and movies. *Behavior Research Methods*, 42, 179–187. doi:10.3758/BRM.42.1.179
- Peterson, M. P. (Ed.). (2003). *Maps and the Internet*. Oxford, UK: Elsevier Science.
- Peterson, M. P. (2015). Evaluating mapping APIs. In: A. Vondrakova, J. Brus & V. Vozenilek (Eds.), *Modern Trends in Cartography: Lecture Notes in Geoinformation and Cartography* (pp. 534). Berlin: Springer.
- Pieters, R. (2008). A review of eye-tracking research in marketing. *Review of Marketing Research*, 4, 123–147.
- Pieters, R., & Wedel, M. (2004). Attention capture and transfer in advertising: Brand, pictorial, and text-size effects. *Journal of Marketing*, 68, 36–50.
- Pirolli, P., Fu, W.-T., Reeder, R., & Card, S. K. (2002, May). *A user-tracing architecture for modeling interaction with the world wide web*. Paper presented at the International Working Conference on Advanced Visual Interfaces, Trento, Italy.
- Ponce, H. R., López, M. J., & Mayer, R. E. (2012). Instructional effectiveness of a computer-supported program for teaching reading comprehension strategies. *Computers and Education*, 59, 1170–1183.
- Poole, A., & Ball, L. J. (2006). Eye tracking in human computer interaction and usability research: Current status and future prospects. In C. Ghaoui (Ed.), *Encyclopedia of human computer interaction* (pp. 211–219). Hershey, PA: Idea Group.
- Popelka, S., & Brychtova, A. (2013). Eye-tracking study on different perception of 2D and 3D terrain visualisation. *Cartographic Journal*, 50, 240–246.
- Rayner, K. (1998). Eye movement in reading and information processing: 20 years of research. *Psychological Bulletin*, 124, 372–422. doi:10.1037/0033-2909.124.3.372
- Rayner, K. (2009). Eye movements and attention in reading, scene perception, and visual search. *Quarterly Journal of Experimental Psychology*, 62, 1457–1506. doi:10.1080/17470210902816461
- Recarte, M. A., & Nunes, L. M. (2000). Effects of verbal and spatial-imagery tasks on eye fixations while driving. *Journal of Experimental Psychology: Applied*, 6, 31–43. doi:10.1037/1076-898X.6.1.31
- Reeder, R. W., Pirolli, P., & Card, S. K. (2001, April). *WebEyeMapper and WebLogger: Tools for analyzing eye tracking data collected in web-use studies*. Paper presented at the CHI '01 Conference on Human Factors in Computing Systems, Seattle, Washington.
- Reichle, E. D., Warren, T., & McConnell, K. (2009). Using E-Z Reader to model the effects of higher level language processing on eye movements during reading. *Psychonomic Bulletin & Review*, 16, 1–21. doi:10.3758/PBR.16.1.1
- Reimer, B., & Sodhi, M. (2006). Detecting eye movements in dynamic environments. *Behavior Research Methods*, 38, 667–682. doi:10.3758/BF03193900
- Rensink, R. A. (2002). Change detection. *Annual Review of Psychology*, 53, 245–277. doi:10.1146/annurev.psych.53.100901.135125
- Rodden, K., & Fu, X. (2007, July). *Exploring how mouse movements relate to eye movements on web search results pages*. Paper presented at the Web Information Seeking and Interaction Workshop at the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands.
- Roth, R. E. (2011). *Interacting with maps: The science and practice of cartographic interaction*. State College, PA: Pennsylvania State University.
- Roth, R. E., & Ross, K. S. (2012). Extending the Google Maps API for event animation mashups. *Cartographic Perspectives*, 64, 21–40.
- Russo, P., Pettit, C., Coltekin, A., Imhof, M., Cox, M., & Bayliss, C. (2014). Understanding soil acidification process using animation and text: An empirical user evaluation with eye tracking. In M. F. Buchroithner, N. Prechtel, D. Burghardt, & the International Cartographic Association (Eds.), *Cartography from pole to pole: Selected contributions to the XXVth International Conference of the ICA, Dresden 2013* (Lecture Notes in Geoinformation and Cartography, pp. 431–448). Berlin, Germany: Springer. doi:10.1007/978-3-642-32618-9\_31
- Shneiderman, B. (1992). *Designing the user interface: Strategies for effective human-computer interaction* (2nd ed.). Reading, MA: Addison-Wesley.
- Simons, D. J., & Ambinder, M. S. (2005). Change blindness—Theory and consequences. *Current Directions in Psychological Science*, 14, 44–48.
- Slocum, T. A., Blok, C., Jiang, B., Koussoulakou, A., Montello, D. R., Fuhrman, S., & Hedley, N. R. (2001). Cognitive and usability issues in geovisualisation. *Cartography and Geographic Information Science*, 28, 61–75.
- Snyder, J. P. (1987). *Map projections—A working manual*. (US Geological Survey Professional Paper, Supt. of Dpcs. No. I 19.16: 1395U.S.). Washington, DC: Government Printing Office.
- Steinke, T. R. (1979). An evaluation of map design and map reading using eye movement recordings. dissertation. University of Kansas. Lawrence.
- Steinke, T. R. (1987). Eye movement studies in cartography and related fields. *Cartographica*, 24(2), 40–73.
- TheLadders. (2012). *Keeping an eye on recruiter behavior: New study clarifies recruiter decision-making* (Technical Report). Retrieved from TheLadders.com
- Townsend, C., & Kahn, B. E. (2014). The “visual preference heuristic”: The influence of visual versus verbal depiction on assortment processing, perceived variety, and choice overload. *Journal of Consumer Research*, 40, 993–1015. doi:10.1086/673521
- Vaeyens, R., Lenoir, M., Williams, A. M., & Philippaerts, R. M. (2007). Mechanisms underpinning successful decision making in skilled youth soccer players: An analysis of visual search behaviors. *Journal of Motor Behavior*, 39, 395–408.
- Van Assche, E., Drieghe, D., Duyck, W., Welvaert, M., & Hartsuiker, R. J. (2011). The influence of semantic constraints on bilingual word recognition during sentence reading. *Journal of Memory and Language*, 64, 88–107.
- Van der Haegen, L., Cai, Q., Stevens, M. A., & Brysbaert, M. (2013). Interhemispheric communication influences reading behavior. *Journal of Cognitive Neuroscience*, 25, 1442–1452.
- van Drunen, A., van den Broek, E. L., Spink, A. J., & Heffelaar, T. (2009). Exploring workload and attention measurements with uLog mouse data. *Behavior Research Methods*, 41, 868–875. doi:10.3758/BRM.41.3.868



- van Elzakker, C. P. J. M., & Griffin, A. L. (2013). Focus on geoinformation users: Cognitive and use/user issues in contemporary cartography. *GIM International*, *27*, 20–23.
- Vansteenkiste, P., Cardon, G., D'Hondt, E., Philippaerts, R., & Lenoir, M. (2013a). The visual control of bicycle steering: The effects of speed and path width. *Accident; Analysis and Prevention*, *51*, 222–227.
- Vansteenkiste, P., Vaeyens, R., Zeuwts, L., Philippaerts, R., & Lenoir, M. (2014) Cue usage in volleyball: A time course comparison of elite, intermediate and novice female players. *Biology of Sport*, *31*(4), 295–302
- Vansteenkiste, P., Zeuwts, L., Cardon, G., & Lenoir, M. (2013, August). *A hazard perception test for cycling children: an exploratory study*. Paper presented at the 2013 Conference on Eye Tracking South Africa, Cape Town, South Africa.
- Virrantaus, K., Fairbairn, D., & Kraak, M. J. (2009). ICA Research Agenda on Cartography and GIScience. *Cartography and Geographic Information Science*, *36*, 209–222.
- Wedel, M., & Pieters, R. (2006). Eye tracking for visual marketing. *Foundations Trends in Marketing*, *1*, 231–320.
- Wengelin, A., Torrance, M., Holmqvist, K., Simpson, S., Galbraith, D., Johansson, V., & Johansson, R. (2009). Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. *Behavior Research Methods*, *41*, 337–351. doi:10.3758/BRM.41.2.337
- Wilkening, J., & Fabrikant, S. I. (2013). How users interact with a 3D geo-browser under time pressure. *Cartography and Geographic Information Science*, *40*, 40–52.