Supporting Web analytics by aggregating user interaction data from heterogeneous devices using viewport-DOM based heat maps

Supporting Web Analytics by Aggregating User Interaction Data from Heterogeneous Devices using Viewport-DOM based Heat Maps

Fabrizio Lamberti, *Senior Member, IEEE,* Gianluca Paravati, *Senior Member, IEEE,*
Valentina Gatteschi, and Alberto Cannavò

Abstract—The players of the digital industry look at network Big Data as an incredible source of revenues, which can allow them to design products, services and market strategies ever more tailored to users' interests and needs. This is the case of data collected by Web analytics tools, which describe the way users interact with Web contents and where their attention focuses onto during navigation. Given the complexity of information to analyze, existing tools often make use of visualization strategies to represent data aggregated throughout separate sessions and multiple users. In particular, heat maps are often adopted to study the distribution of mouse activity and identify page regions that are more frequently reached during interaction. Unfortunately, since Web contents are accessed via ever more heterogeneous devices, region-based heat maps cannot be exploited anymore to aggregate data concerning user's attention, since the same Web content may move to another page location or exhibit a different aspect depending on the access device used or the user agent setup. This paper presents the design of a visual analytics framework capable to deal with the above limitation by adopting a data collection approach that combines information about regions displayed with information about page structure. This way, the well-known heat map-based visualization can be produced, where interactions can be aggregated on a per-element basis independently of the specific access configuration. Experimental results showed that the framework succeeds in accurately quantifying user's attention and replicating results obtained by manual processing.

Index Terms—Network Big Data, Web navigation, Web analytics, data visualization, heat maps, user's attention, interaction patterns.

1 Introduction

Last years have been characterized by a dramatic increase in the amount of data generated and transmitted over the Network. Data is either produced by machine-to-machine communications [1] or by users' interactions with online services/applications [2]. Information is growing at an incredibly fast pace, representing a clear example of Big Data [3]. Data is more and more regarded as a key source of revenues by all the major actors of the digital market, being often referred to as the "oil of the 21st century" [4].

A data source whose analysis is considered as being capable to provide extremely valuable outcomes is represented by users' behavior during Web navigation, which is often used for optimizing page layout for usability, detecting the subjective interest in information displayed, quantifying the impact of an advertisement campaign, etc. [5]–[7].

To get insights on users' behaviors, a number of software solutions have been developed, which fall in the category of Web analytics tools and services. These solutions generally rely on a rather standard architecture, in which a small piece of software (known as tag) is embedded in Web pages to be monitored and is responsible for gathering data about user's interactions and sending it to the server for processing.

While most of the tools records mouse clicks (used, e.g., to create navigation or content reports), in some cases, user's activity is tracked by considering also mouse movements, which proved to be able to describe, at least in some scenarios, how user's attention focused on screen regions [8].

When it comes to analyze results, a number of graphics representations are used to display both raw data (e.g., playing back all the interactions of a navigation session) and aggregated data (e.g., through a "one-shot view" of all mouse clicks in all the sessions for a given page) [9]. One of the representations that is almost commonplace in existing tools is the *heat map*, a pictorial depiction of information where numeric values are shown using colors [10]. Most of the existing tools produce heat maps displaying mouse clicks/movements. Other tools provide alternative visualizations (referred to as attention, scroll-reach, relevance or exposure heat maps) meant to describe how a page has been scanned during navigation. This way, it is possible to see which areas of a page are registering more activity or are receiving more attention by the visitors. The latter goal can be achieved by leveraging information about the region of the page that is currently displayed in the browser window. This information is generally referred to as the $viewport$, and can be obtained from Web user agents (UAs).

Traditionally, heat map-based representations have been drafted based on page regions, rather than on actual location or aspect of page contents. The assumption is that contents should have received a certain attention because they were included in a region that was shown to the user for a given amount of time. Unfortunately, the assumption above may not be valid in general. In fact, there is no guarantee that a page is rendered in the same way on all the possible devices and UAs. Moreover, languages like Cascading Style Sheets (CSS) give the Web designer the ability to create so-called responsive layout, controlling how page elements will be displayed on a device with a given orientation, resolution, etc. In these cases, Web analysts are usually provided with per-device heat maps. Only few existing tools draw aggregated heat maps, showing users' interactions with Web pages on multiple devices. Nonetheless, results they provide are often not accurate, especially with responsive layouts.

Although, in some cases, per-devices heat maps could be enough (e.g., to study usability), aggregate heat maps could be helpful should Web analyst want to reason in terms of displayed contents. In fact, independently of the device used, there could be some page elements (like images, text containing relevant information, etc.) which could capture, more than others, users' attention. By looking at an aggregate heat map, Web analyst could then decide to move such elements to the top of the page to increase conversion rate.

This paper presents an approach for aggregating data about user's attention on Web pages independently of the access device/UA used and for visually representing them in a single heat map. The idea, originally presented in [11], is to exploit data about viewport changes and combine it with information about page structure that can be obtained from the Document Object Model, or DOM. Hence, the devised framework and visualization method will be later referred to as VDHM, i.e.,"Viewport-DOM based Heat Map".

Being based on information extracted from the DOM, the VDHM approach is able to provide more accurate results compared to existing solutions, especially with responsive layouts. With respect to [11], the devised framework is able to deal also with "dynamic" modifications to page structure and elements appearance that could be, e.g., activated by client-side scripts or generated by user's interactions with page content. Moreover, in [11] the validity of the proposed approach was only proved by working on a dummy Web page under controlled conditions. In this paper, a number of experiments have been carried out to accurately quantify effectiveness. Users have been asked to use both desktop and mobile devices to access different pages, each characterized by diverse responsive behaviors, and obtained results have been compared with a state of the art Web analytics tool showing better performances.

2 Background

In the last decade, visual analytics has become a key tool for decision making in industry [12]. One of the tools that has proved to be particularly effective, especially for Big Data visual analytics, is the heat map, due to its ability to convey an immediate representation of aggregated multi-dimensional and dynamic information [13].

Some some pioneer research activities [14], [15] proposed to use heat maps also for Web analytics, and combined them with gaze tracking to analyze users' attention on different devices. More recently, a number of research works and (commercial) platforms, like ClickTale (www.clicktale.com), Open Web Analytics - OWA (www.openwebanalytics.com), etc. relied on heat maps to analyze user's behavior during Web navigation, by using colors to indicate the amount of activity concentrated in a given page region [7]. In this context, aggregation capabilities of heat maps play a key role, as they could be used to display in a single picture data pertaining interactions with the same page collected over multiple navigation sessions with different devices and UAs. Information shown in a heat map can be used, e.g., to evaluate the visual appeal of a page, assess user experience, reorganize contents for usability, etc. [5], [6], [9], [16].

With the spread of mobile devices, Web analytics approaches were modified to consider touch-based interactions, by basically mapping clicks onto touches. Thus, tools like Crazy Egg (www.crazyegg.com), MouseFlow (mouseflow.com), Hotjar (www.hotjar.com), etc. were designed to record tap, zoom, pinch, scroll and swipe gestures, which can then be displayed in a heat map. However, on small-screen devices a touch event might not always indicate a page region that is really of interest for the user. For instance, on a smartphone, the user may touch the screen to scroll and zoom the page, and fingers may be intentionally moved outside the region containing information of interest, to not occlude it [17]. Thus, a mouse-based heat map would hardly keep its meaning when used to study users' activity on mobile devices. To address this issue, several works [17]–[19] proposed to use viewport changes to obtain information about user's attention on mobile and touch-enabled devices. In particular, when a swipe gesture is performed, the change in viewport position is regarded as an indicator of a skipping or reading behavior [18]. When a zoom is performed, the change in both viewport position and size indicates an increase in user's attention for the region considered [19].

Using the viewport as a proxy of user's interest on a page region is not novel. Several commercial Web analytics solutions already gather viewport data and represent them in the so-called scroll-reach or attention heat maps. Such heat maps complement click- and movement-based heat maps, by depicting with warmer colors those regions that have been shown for a longer time on the user's screen. Despite their common usage, the first research works proposing viewport-based heat maps for mobile devices considering swipe and zoom gestures were published only in 2012 [17], [20]. The authors of these works underlined that, for investigating usability, aggregation of touch data collected on devices with different screens is not useful, as screen size significantly influences interaction patterns [21]. Hence, they only aggregated data for devices with the same screen size.

Despite limitations observed in usability studies, when heat maps are to be used for analyzing user's attention, their aggregation capabilities assume a fundamental importance, since they allow Web analysts to get the whole picture about how page information is being accessed on different devices, UAs, etc. Sadly, in some cases, aggregation could be hard to be achieved, e.g., when responsive Web pages are used. Responsive pages are designed to adapt the visual appearance of their elements based on access configuration. Hence, aggregating viewport-based data collected with different devices could lead to inconsistent graphics representations.

When visualization is not necessary, a way that has been pursued to aggregate Web navigation data gathered on heterogeneous devices consists in focusing on page elements, rather than on page regions. Specifically, page DOM can be queried to retrieve information about elements position, size, etc. This information could then be exploited by Web analytics tools to generate statistics on page navigation, (e.g., about how elements are clicked or tapped, links are activated, etc.) or to reproduce individual user's interactions.

TABLE 1
Features provided by heat map-based Web analytics tools.

	Heat map types	Resp. layout	Dynam. modif.	Devices aggreg.	Sess. playback
Crazy Egg	CSA	x	x	x	
Clicktale	CSMA	x	x		x
Hotjar	CSM	x	x		x
OWA	C	x			x
MouseFlow	CSMA	x	x	x	x

When visualization is needed, a number of commercial solutions could be used, depending on Web analysts' needs. Table 1 summarizes the main features of the most known heat map-based Web analytics tools. Information has been collected on companies' Websites or through interaction with tools' demo (when available). In particular, tools are compared in terms of supported heat map types (recording mouse clicks C, scrolls S, movements M, and attention A), of their ability to deal with responsive layouts (on different devices) as well as with dynamic page modifications, of their capability to draw heat maps by aggregating information collected on different devices (desktop, tablet, smartphone), and of the ability to playback single sessions.

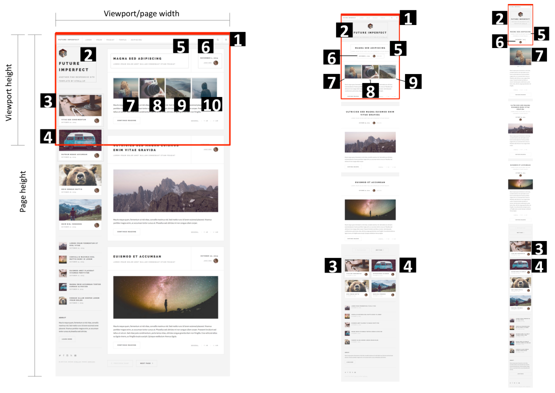
According to Table 1, there are only two tools, Crazy Egg and MouseFlow, that focus on user's attention and provide an aggregate view of information displayed on different devices. Nonetheless, tests performed on publicly-available demo versions showed that results provided by these tools are not always correct, especially with responsive layouts. Moreover, during zoom interactions, both the tools only track horizontal changes in the viewport, making impossible to determine, in cases of multiple elements per line, which was the element actually of interest for the user.

To address the above issues, this work proposes an approach based on viewport and DOM information for aggregating navigation data. The basic idea was presented in [11], but the original framework was not able to deal with dynamic layouts, and its functioning was tested only on a dummy Web page without comparing it with existing tools. In this work, a methodology for dealing with changes that could occur to page structure during navigation is introduced. Moreover, the results of extensive tests, performed on "real" responsive and non-responsive Web pages, and of a comparison with a state of the art platform are reported.

3 Basic Idea and Novel Contribution

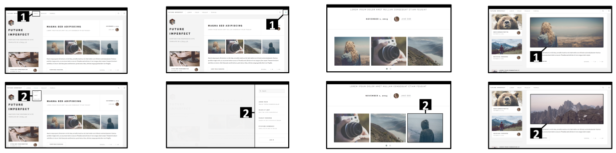
In the following, the idea behind the designed framework as well as the novelty of the proposed approach will be presented, by making reference to the sample Web page shown in Fig. 1 and Fig. 2 (www.vdhm.altervista.org/pages/fi/index.html). The page was created starting from a free responsive template (www.html5up.net/future-imperfect) and modified to present also some common dynamic features.

In particular, in Fig. 1a the aspect of page elements (labeled from 1 to 10) for a desktop device is shown. Fig. 1b and Fig. 1c illustrate changes that may occur to the above elements when a different size for the UA window (on tablet and smartphone devices, respectively) is used for accessing the page (or, similarly, when the window is resized).



(a) (b) (c)

Fig. 1. Aspect changes for a Web page accessed with different devices desktop (a), tablet (b), smartphone (c).



(a) (b) (c) (d)

Fig. 2. Dynamic changes activated by user action, timers, etc.

The title on the top left (2) of the desktop view is moved to the center on the other two devices. A similar behavior characterizes elements 5 and 6, which are arranged in a single column structure, and centered on smaller devices. The menu and the search icon in the navigation bar (1), which are fully displayed in the desktop page, show less items on tablet devices, and are removed on smartphones. Elements 3, 4 in the left column are moved towards the bottom the page and rearranged in a 2 × 2 layout (tablet), or vertically (smartphone). Finally, only three out of the four images shown on desktop devices (7–10) are displayed on tablets (only one on smartphone).

When a heat map is produced by aggregating viewport data from Fig. 1a–c with available tools, results would not correctly reflect users' behaviors, since elements within the page region actually contained in the viewport, i.e., in the UA window, are not the same (because they have been eliminated from the layout, hidden, moved, etc.). Things would change if page elements, rather than regions, are considered. In fact, it would be quite easy to know whether the user can actually see a particular element or not (and how much of it, as well as how long). Once this information is available for any device/UA used, it could be aggregated to create a single heat map, where elements contribution is calculated based on the amount of time each element was displayed to the user in every different configuration.

In [11], a preliminary approach was proposed to exploit information about elements position, size, etc. that can be extracted from the DOM, e.g., right after page load. In particular, after every scroll and zoom interaction, elements position and size obtained at page load is checked against the viewport, in oder to determine whether they are still visible or not. The approach in [11], however, did not consider those element that are natively meant to change their appearance either automatically as time passes or in response to user's actions, like drop-down lists, dialog windows, scrollable contents, etc. In fact, when such "dynamic" elements are used, information in the DOM may vary, even after page load, while the user remains on the same page.

To get an overview of possible dynamic behaviors, it could be useful to consider the examples presented in Fig. 2 (where the upper part depicts the appearance at page load, and the lower part the occurred changes). In Fig. 2a, a dropdown menu is shown. The initial size of the menu (labelled with 2) is null. When the mouse is moved over the visible text element (1), the page responds to the `onmouseover` event and makes the above element appear. A similar situation is depicted in Fig. 2b, where changes in the DOM structure are triggered by the `onclick` event on the element (1), which opens a sliding menu (2). Fig. 2c shows a carousel endowed with control widgets. When the user clicks on one of the controls, the panel hosting the images is shifted to the left (e.g., by acting on the CSS `transform` property), and the image to the very right (2) becomes visible. Lastly, a common scenario where two DOM elements are switched, is depicted in Fig. 2d. At page load, the element (1) is shown. After some time (or in response to a user-generated event), the DOM is modified and another element (2) is displayed.

Behaviors shown in Fig. 2 are quite common in modern Web pages and profoundly modify the aspect of a page during navigation. Hence, for Web analytic tools it becomes more and more important to deal with such behaviors, in order to avoid providing inconsistent results.

In this view, the approach presented in this work aims at taking a step forward with respect to [11], where the analysis is performed only on information obtained at page load. Here, data regarding dynamic changes in the DOM are also considered, by performing multiple renderings of the page and acquiring information on the visibility of elements of interest and their containers. Furthermore, in [11] the approach was presented by referring to a dummy Web page, without performing any test to verify the "quality" of aggregation. Here, an in-depth evaluation of produced heat maps based on real cases is provided, with the aim to quantitatively confirm the validity of the proposed approach.

4 Proposed Framework

The VDHM framework, shown in Fig. 3, is made up of three components: a) a script running on the client side called the $Reporter$, which is in charge of gathering information about user's activity and its effects on page structure, b) a server-side module, referred to as the $Service$, handling client requests and storing interaction data in a relational database, c) an application supporting configuration and analysis tasks named the $Manager$, which is used to configure the Reporter module by selecting page elements to be monitored, and to produce graphics reports based on information collected during navigation. In the following, implementation details be discussed by making reference to the sample Web page shown in Fig. 1 and Fig. 2.

4.1 Reporter

To gather data about user's attention on page elements independently of the device/UA used, the Reporter dialogues with the UA and page DOM, collects information at page load or during user's interaction and periodically transmits data to the server side via asynchronous AJAX calls.

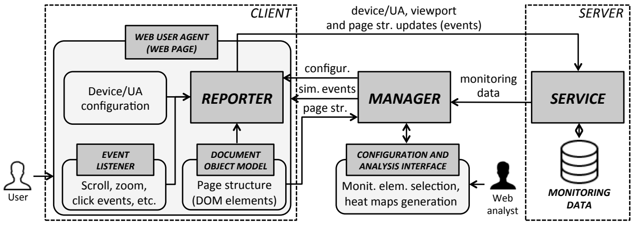


Fig. 3. High-level architecture of the VDHM framework.

When page is loaded, the device/UA the script is executed into is identified (to let the Manager subsequently produce per-page and per-device/UA graphics reports). Then, viewport size and location (in terms of width and height, plus top and left coordinates), as well as the initial page structure (as a collection of bounding boxes and visibility information for all the DOM elements of interest) are determined. During user's activity, the Reporter tracks changes to the above information (e.g., page scrolls, mouse movements on elements, etc.) and updates the Service accordingly.

Script functionalities are implemented by means of six sets of ad hoc functions, which are described in Table 2.

Reporter and Service components interact as follows (more details could be found in [11]). When the script is activated, the Reporter sends a message to inform the Service about the device/UA configuration. The Service performs two operations: a) it extracts the UA id (`uid`) and stores it to later aggregate on the basis of the used device, b) it sends back to the client a session id (`sid`), which will be included by the client in all the session messages exchanged and will be used by the Service to identify the data collected on a specific device/UA and IP address. The Reporter will then send, at predefined intervals, user's interactions, page modifications and a timestamp in milliseconds (`ts`), unless a given payload size is reached (in this case transfer is started immediately). Timestamps will be used by the Manager to handle interactions and modifications in the right order.

4.2 Service

The Service component is written in PHP and performs two tasks: a) it accepts Reporter's AJAX requests, extracts and stores monitoring data, b) it delivers to the Manager application information required to produce the visual reports.

The underlying repository is a MySQL database, with a schema based on three tables, namely `session`, `event` and `dom`. The `session` table contains generated `sid`s and `uid`s, and the timestamp sent by the Reporter. The `event` table contains page and viewport details. Rows in this table are distinguished by a letter, which refers to the specific event described (in this case, `p` - page - and `v` - viewport), and contain information on page/viewport width and height (`w`, `h`), and on viewport horizontal and vertical position (`x`, `y`). The `dom` table contains information about DOM elements received in type `d` messages (in terms of window width `w` and height `h`, coordinates of the top-left corner of element's bounding box `x`, `y`, and element's visibility

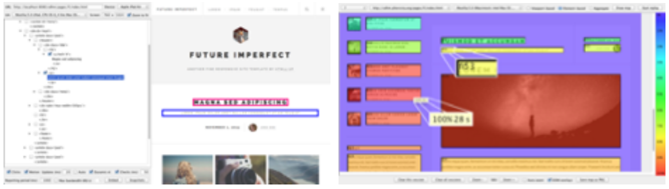
TABLE 2
Sets of functions developed for the Reporter component

#	Purpose	Function(s)	Description
1	Get information on device's UA, page URL	`getUserAgentInfo()`	Gets details about the UA by accessing the `userAgent` property of the `navigator` object (which contains the user-agent header sent to the server, and includes information about the current device).
2	Get page and viewport configuration	`getPageSize()`, `getViewportInfo()`	Obtain page and viewport information. Since different UAs store location and size data in different ways, these functions consider various DOM elements and their properties in order to guarantee cross-browser compatibility.

3	Get elements position, size and visibility	`getDom()`
		`getElementInfo()`
			Gets information for the DOM elements to be monitored (identified by a particular attribute that can be added to page elements in the configuration stage by means of the Manager application).
			Gathers type, size, coordinates of element's bounding box and visibility information.

4	Track users' interactions	`getEvents()`	Tracks page events. It starts/stops page monitoring based on `onload` and `onbeforeunload` events and monitors scroll/resize, and click/touches events.
5	Monitor changes to page structure and elements appearance	`check DOM()`	Checks the DOM for changes in position/size/visibility of elements of interest. It is invoked when an event is fired, or at fixed intervals, configured through the Manager application (using the `setInterval()` function of the `window` object). Examples of tracked events are `mouseover` or `mouseout` (to get modifications to elements underneath the mouse cursor), or `mousedown`, `mouseup` and click/touch events (to get changes occurred on elements of interest).
6	Send data to the Service	`sendData()`	Generates a string containing the type of the tracked interaction/recorded changes, and send it to the Service using an AJAX request.

`visib`). A timestamp is used to track changes to elements position, size and visibility over time. Click/touch and mouse move interactions are recorded in the `event` table as type `i` rows containing also the `x` and `y` coordinates of the click/touch. Scroll, swipe/drag/flick and zoom operations are all recorded as type `v` rows in the `event` table (as they correspond to viewport changes), and contain `x`, `y`, `w`, `h` information. The same considerations apply to messages exchanged when the page is hidden, re-displayed or unloaded. When the UA window is resized, the `event` table is updated to store type `r` messages containing page/viewport `w`, `h`, whereas the `dom` table records elements changes.



(a) (b)

Fig. 4. Manager application interface for (a) configuration and (b) reporting. Element bounding boxes and identifiers can be displayed in overlay, and attention (in seconds) obtained by clicking on the map.

4.3 Manager

The Manager component has been developed as a Web application, built around a Java applet responsible for communicating with the Service and carrying out visualization tasks requested by the analyst.

In a preliminary configuration phase, the analyst can exploit the Manager to explore a Web page and choose the elements to be monitored. Fig. 4a shows the Manager configuration interface for the sample page considered so far. The Applet uses the JSoup library (jsoup.org) to parse the page DOM and to create a tree-based visualization of it (on the left). Meanwhile, it renders the page using the JxBrowser library (www.teamdev.com/jxbrowser) (on the right). The analyst can specify page size and UA characteristics to mimic page appearance on different devices/UAs. When an element is selected in the tree, its bounding box is displayed in overlay in the page representation. Elements of interest are marked with a specific attribute (namely `vid`), used by the Reporter to determine the elements to be analyzed and to identify them in the type `d` messages. The Manager also lets the analyst define some parameters controlling the script, like update frequency, bandwidth usage, etc. After configuration, the Manager embeds the script in the page and creates a snapshot of it, which will be used as background image for generated heat maps.

A single snapshot per device/UA gathered right after page load would not show dynamic changes occurring over time, e.g., due to user's actions on the page. To manage this issue, w.r.t. [11] a new methodology has been developed, in which multiple renderings including changing elements are created during configuration by simulating user interaction and observing page reactions. Mouse is programmatically moved in the browser-based component over all the elements in the DOM. Click/drag/scroll events are then simulated without leaving the page, and possible changes to elements visibility are monitored. When a hidden element becomes visible, a new snapshot of the page is gathered, and the portion of the image corresponding to element bounding box is recorded. The `vid` of changed element is associated with the image, so that its bounding box will be displayed in the right place during heat map creation. When the page has been processed, image portions extracted are assembled into one or more background images, which will be used to display heat map-based attention data for associated elements. The background images created for the sample page (specifically, for a tablet device) are shown in Fig. 5a–c.

When used for reporting, the interface of the Manager application changes (Fig. 4b) to present to the analyst a list of pages for which monitoring data have been collected. Once a page is selected, two visualization modes are available, i.e., *viewport-based* (the one traditionally available in existing tools and shown in Fig. 5d–f) and *element-based* (the novel proposed in this paper, shown in Fig. 5g–i).

In the viewport-based visualization mode, the analyst can choose whether to examine a specific session for the selected page or aggregate multiple sessions together. In the latter case, only sessions characterized by the same `uid`, i.e., same device/UA characteristics, will be aggregated. To create the heat map, an empty matrix is first allocated and initialized. Then, a query is performed on the database through the Service to gather relevant entries from the `event` table, starting from page load. If a change in viewport position or size is found, time difference w.r.t. page load timestamp is computed and accumulated in the matrix for pixels involved. When all the entries have been considered, matrix values are normalized in the [0,1] interval and mapped onto the Hue component of a reverted HSV color scale (in the range from 0.0 to 0.7). That is, page regions that have been longer visible are assigned a red color (Hue equal to 0.0), whereas regions that rarely fell in the viewport are displayed in blue. Finally, the matrix is overlapped to the background image(s) and the heat map displayed.

As said, with the viewport-based visualization, an aggregation of information gathered under different device/UA configuration conditions is not feasible, since elements contained in a given page region might not remain the same.

In the proposed element-based visualization mode, the above limitation is addressed by centering heat map creation around the information collected in the `dom` table. As before, the analysis begins with the selection of a page. For each element visible in the page for a particular device/UA, the amount of area included in the viewport at any given time is determined by comparing its bounding box with type `v` entries in the `event` table. For each entry, the ratio between visible and total element area is multiplied by the time the area was displayed to the user and the result accumulated. Once all elements and entries have been processed, collected values are normalized and used to generate the heat map. In this case, HSV color values are applied to element bounding boxes rather than to viewport regions.

Compared to existing tools, the advantage of the visualization mode enabled by the VDHM framework is that the analyst can now choose to aggregate data about user's attention collected with different devices/UAs in a single graphics representation. The heat map aggregating data for the three devices/sessions considered is shown in Fig. 5j.

5 Experimental Results

In this section, a set of experiments carried out on pages selected from several public websites and aimed to assess the effectiveness of the devised methodology will be presented.

Experiments were focused on four different Web pages (referred to as $Library$, $Tourism$, $Accommodation$ and $News$), chosen to exemplify different design and interaction patterns. Two pages were characterized by a non-responsive layout, whereas the visualization of the other two pages changed significantly depending on the resolution (precisely, the window width) of the access device/UA considered. Tests were performed on different devices, namely a MacBook Air, an iPad Air and a LG Nexus 5 (with a horizontal resolution of 1440, 768 and 370 pixels, respectively). Device features had an impact not only on the way page elements were organized on screen, but also on user's behavior during navigation. In fact, with non-responsive layouts, page navigation required the use of both horizontal and vertical scrolling as well as zoom in and out operations.

The first page (www.vdhm.altervista.org/pages/library/index.html) is the HTML/CSS version of a research paper accessed through IEEE Xplore. Here, information is mostly concentrated in the central column, requiring the user to scan the page through vertical scroll operations. The second page (www.vdhm.altervista.org/pages/tourism/index.html) contains tourism information for the Valle d'Aosta Region, Italy. Information depicted is distributed in diverse locations, both along the horizontal and vertical directions. Thus, interaction is expected to be more complex. The third page (www.vdhm.altervista.org/pages/accommodation/index.html) is the result of a query on the House Trip lodging website. Information organization is quite simple and schematic, with elements changing position and, in some cases, visibility (e.g., for lower resolutions), depending on the device used. The fourth page (www.vdhm.altervista.org/pages/news/index.html) is the homepage of the BBC website, and was selected to show a combination of a responsive layout with a more sophisticated distribution of information. The page represents a challenging scenario. In fact, its appearance changes significantly when passing from one device to another, since the multi-column layout rearranges content by moving elements to different parts of the page.

5.1 Methodology

In order to assess the validity of the VDHM methodology, two evaluations were performed. The first one, referred to as $accuracy$ evaluation, aimed at proving that the VDHM framework is able to collect accurate information about page elements that were longer displayed to the users during navigation sessions involving heterogeneous devices. Results obtained using the VDHM approach were compared with those produced by MouseFlow, a commercial solution supporting aggregate heat maps (later referred to as "MF"). As it will be discussed in the following, accuracy evaluation showed that heat maps drawn with MF were not able to correctly reflect the behavior of users during navigation. Hence, in the second evaluation, named $coherency$ evaluation, instead of comparing the heat maps produced by the two tools, the "quality" of VDHM aggregated heat maps was evaluated by making reference to results obtained with manual processing. The objective was to check if the devised automatic aggregation mechanism could replicate manual processing tasks, in which Web nalysts use existing tools to draw conclusions about attention received by page elements, independently of the access devices used.

For each page, three tasks were created, requesting the user to search for specific information in the page and use it for answering a set of questions (designed to focus user's attention on an exact set of elements). Table 3 reports questions defined for each task and page elements (identified by the `vid` attribute assigned in the configuration step) containing information required for answering them. 30 volunteers were selected among university students and equally assigned to the three devices. During the test, each user was allowed to freely navigate the four pages and carry

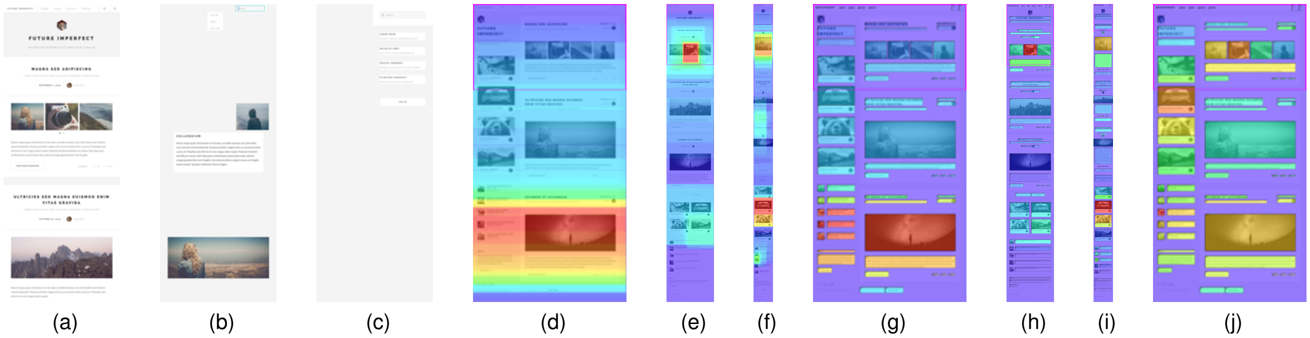


Fig. 5. Heat map background images recording dynamic behaviors (a–c), and generated heat maps: (d–f) viewport-based, (g–i) element-based on desktop, tablet and smartphone, and (j) aggregated (VDHM).

TABLE 3
Tasks and corresponding element identifiers for the first test phase.

Case study	Task id.	Elem. id.	Task (Question)
Library (non responsive)	L.1	4, 5 6, 7	Four system requirements in the section on authors' contribution
	L.2	42 43	User interface components in the figure on Search User Interface
	L.3	64	Locations where the first listed author obtained B.Sc./M.Sc.
Tourism (non responsive)	T.1	37, 38	Hotel booking phone number
	T.2	25 26 27, 28	Cost for one night with option "The Mont Blanc Experience" and destination Mont-Blanc cableway
	T.3	6-7	Open period of the ski resort
Accommod. (responsive)	A.1	43 44	Cost per night of the apartment with four bedrooms
	A.2	47-48	Most expensive apartment
	A.3	17 18	Cost per night of the apartment with green walls
News (responsive)	N.1	124	Today's stock market indexes
	N.2	46, 47	Title of the article about robotics
	N.3	67, 68	Title of the article about athletics

out the tasks by using only one device (to avoid learning). For each page, the VDHM framework created a heat map specific for each device used, by aggregating data collected from all the navigation sessions on that device, as well as an aggregated heat map, based on all the whole set of sessions.

Accuracy was then evaluated by analyzing whether the rank computed by the VDHM and MF methods was able to reflect users' "guided" navigation. In particular, the amount of time spent by the users on given page areas was extracted from the heap maps created by the two tools and compared.

Once studied whether aggregate heat maps could be able to represent users' behavior, coherency evaluation was carried out. The goal was to check if the approach used for aggregating data could mimic human reasoning. Hence, VDHM results were compared with those that could be obtained by manual processing on data that would be made available by common tools for Web analytics. Data collected by the 30 volunteers were used to produce conventional viewport-based heat maps, one per each page and device (thus simulating the behavior of existing tools).

TABLE 4
Ranking of elements to be considered for each case study.

Library			Tourism			Accommod.			News		
($F = 7$ elem.)			($F = 8$ elem.)			($F = 6$ elem.)			($F = 5$ elem.)		
El.	Rank		El.	Rank		El.	Rank		El.	Rank	
id.	VDHM	MF	id.	VDHM	MF	id.	VDHM	MF	id.	VDHM	MF
4	2	14	6	5	9	17	11	34	46	4	47
5	1	17	7	1	1	18	9	32	47	5	24
6	3	16	25	29	30	43	5	5	67	2	5
7	6	7	26	24	24	44	1	3	68	1	13
42	5	20	27	27	31	47	2	1	124	6	36
43	4	10	28	30	33	48	7	11			
64	9	8	37	6	2						
			38	7	5						

Then, two human evaluators selected among university staff (unaware of the specific tasks assigned in the first phase, to avoid bias) were asked to work on these maps. They had to manually (i.e., visually) combine information contained therein to sort out page elements based on the attention (expressed in seconds) they thought each element had actually received (computed by averaging values displayed in the maps). Processing was carried out with the Manager application by superimposing the bounding boxes of monitored elements, thus letting evaluators coherently compare the same graphics elements in all the visualizations (Fig. 4b). By clicking on the map, a per-pixel measure of the average time spent during navigation on the various page regions could be obtained. Performances of the proposed approach were assessed by computing the correlation between automatically- and manually-obtained results.

5.2 Accuracy Evaluation

Fig. 6 shows the VDHM element-based aggregated heat maps depicting navigation data for the four case studies (viewport-based heat maps used in the second phase are included to enable comparison). High quality versions are available online at www.vdhm.altervista.org/vdhmmaps.

Heat maps drawn with MF could be found at www.vdhm.altervista.org/mfmaps. MF provides two numeric values, namely an average viewing time and an "engagement" time taking into account users' interactions with page elements. Since here the objective was to compute the accuracy in terms of viewing time, the first value was used. The amount of seconds an element was shown to the user was computed by averaging viewing time for its top, center and bottom. It is worth remarking that MF heat maps are

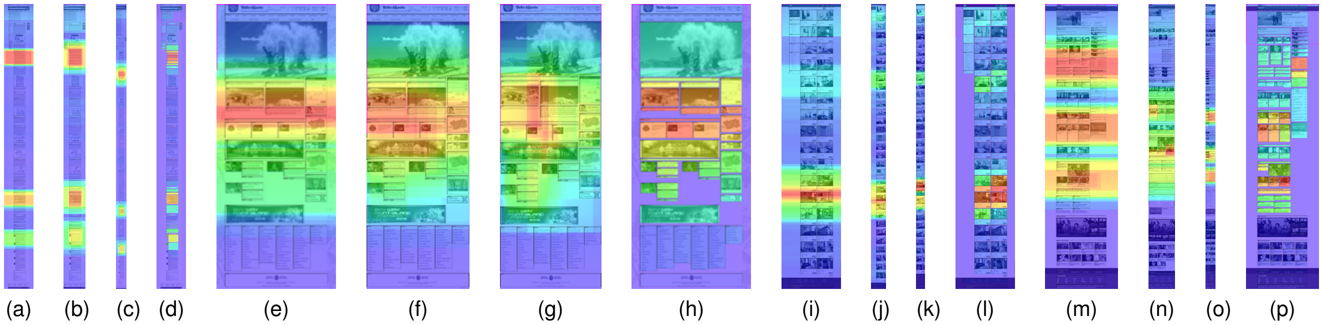


Fig. 6. Heat maps with overlays for the four case studies: (a-d) Library, (e-h) Tourism, (i-l) Accommodation, and (m-p) News. For each case study, three viewport-based maps (obtained by aggregating navigation sessions on the desktop, tablet and smartphone device, respectively) and one element-based map (obtained by aggregating all the navigation sessions, independently of the access device used) are shown.

drawn based on the engagement time; hence, in some cases, the reader might not find a direct correspondence between results reported in this Section and in the heat maps.

For each VDHM and MF map, elements can be ranked based on the attention they received (i.e., display time), overall, in the various navigation sessions with the three devices. Ranking results are illustrated in Table 4. For each case study, the number of elements that, according to Table 3, users were expected to focus onto (F) is reported. For each element (`vid`), the rank computed by VDHM and MF approaches is given. Maximum accuracy would correspond to having, for each element, a rank lower or equal to F. This way, elements with the highest rank would correspond to those the users should have focused their attention onto. By expressing data in Table 4 in a format easier to read, e.g., as deciles (10-quantiles), it could be observed that most of the elements identified by the VDHM method fall in first two deciles, confirming a generally high accuracy.

However, during navigation, there could be page elements that were visible in the viewport, though user's attention was focused on other elements (e.g., on devices with large screens and/or resolutions). In this case, ranking results may be negatively affected. An example of such behavior is provided by the Library case study. Here, ranks for the elements to be considered should be lower or equal to 7. This is generally true, except for element with `vid` equal to 64 (containing authors' information), whose rank is 9. This is due to the fact that the particular element is rather small w.r.t. window size and close to other monitored elements. A slightly more critical situation is represented by the Tourism case study, for which half of the elements fall between the 6-th and the 7-th decile. Motivations for these results can be found by considering the page structure, which is characterized by a non-responsive layout and a complicated content organization. When accessing the page, e.g., from a tablet device, the users had to focus on elements located on both the left and right side, in order to carry out the task. However, central regions of the page were displayed at all times. As it can be seen in Fig. 6g, this produced a warm area in the middle of the heat map, corresponding to the overlap between left and right scrolls.

With regard to the MF rank, the overall accuracy is lower than the VDHM one. Interestingly, when deciles are considered, MF seems to have a behavior similar to VDHM with non-responsive pages (Library results fall in the first two deciles, whereas Tourism results confirm the ones obtained by VDHM). However, in the Accommodation case study, 1/3 of results fall between the 4-th and the 5-th decile, whereas for News case study, the majority of results fall between the 2-nd and the 5-th decile, thus underlying an incorrect aggregation of viewing time. This aspect confirms the validity of the proposed DOM-based approach with respect to approaches based only on viewport information.

5.3 Coherency Evaluation

As said, this phase was aimed at studying the correlation between manually-obtained and VDHM results.

In the experimental setup considered, evaluators manually examined three heat maps (one per device) per case study, each containing a number of elements of interest (N) between 44 and 145 (defined by their bounding box in overlay, and labeled with the `vid` identifier).

Results in terms of Pearson's and Spearman's correlation coefficients for both the human measurements (H.M.) and the VDHM framework are reported in Table 5. Coefficients were computed by organizing elements in deciles based on ranks resulting from manual processing and considering elements in the top-k% together. For instance, $k = 0.5$ means that all the elements in the top 50% of the ranking were considered. Coefficients for H.M. measure the correlation between results obtained by the two evaluators. Coefficients for VDHM provide an indication of the relation between measurements produced by the designed method and measurements resulting by averaging manually-obtained times. Pearson's r values are computed on raw data, whereas Spearman's r_S values are calculated on rankings.

Not surprisingly, for smaller values of k, results are less correlated. However, for both coefficients, H.M. always show a strong correlation ($r > 0.9$ and $r_S > 0.8$). Hence, results produced by evaluators through manual processing can be treated as a reference, and it makes sense to compare their correlation with automatically-generated measurements. On average, correlation values for VDHM are only slightly lower than for H.M. results. Worst cases are found by considering the first decile (i.e., $k = 10\%$), where $r = 0.722$ (Tourism) and $r_S = 0.666$ (Accommodation).

Moreover, it could be seen that, in some cases, correlation decreases from $k = 100\%$ to $k = 50\%$, and, subsequently, increases again in the range ($30\% \leq k \leq 40\%$). This could

TABLE 5
Pearson's and Spearman's correlation between human measurements (H.M.) and between human- and VDHM-obtained results.

k	Library ($N = 104$ el. of interest)				Tourism ($N = 44$ el. of interest)				Accommodation ($N = 80$ el. of interest)				News ($N = 145$ el. of interest)			
[%]	Pear.'s r		Spear.'s r_S		Pear.'s r		Spear.'s r_S		Pear.'s r		Spear.'s r_S		Pear.'s r		Spear.'s r_S	
	H.M.	VDHM	H.M.	VDHM	H.M.	VDHM	H.M.	VDHM	H.M.	VDHM	H.S.	VDHM	H.M.	VDHM.	H.M.	VDHM
100	0.998	0.990	0.995	0.983	0.996	0.996	0.993	0.992	0.996	0.993	0.984	0.980	0.994	0.965	0.996	0.956
90	0.998	0.990	0.993	0.978	0.996	0.996	0.992	0.992	0.998	0.993	0.988	0.973	0.993	0.961	0.995	0.956
80	0.998	0.989	0.990	0.972	0.989	0.989	0.987	0.991	0.997	0.992	0.984	0.963	0.992	0.956	0.993	0.946
70	0.998	0.988	0.988	0.969	0.982	0.990	0.980	0.986	0.997	0.992	0.983	0.945	0.990	0.945	0.990	0.925
60	0.998	0.987	0.982	0.952	0.984	0.986	0.975	0.979	0.997	0.991	0.990	0.933	0.989	0.932	0.988	0.894
50	0.997	0.985	0.972	0.966	0.982	0.980	0.963	0.967	0.997	0.992	0.989	0.940	0.988	0.921	0.988	0.875
40	0.997	0.987	0.984	0.983	0.972	0.966	0.961	0.936	0.997	0.991	0.991	0.983	0.984	0.902	0.992	0.886
30	0.996	0.984	0.967	0.962	0.956	0.965	0.973	0.929	0.996	0.990	0.981	0.968	0.972	0.950	0.981	0.948
20	0.997	0.982	0.993	0.976	0.919	0.813	0.905	0.714	0.994	0.986	0.968	0.924	0.936	0.904	0.957	0.925
10	0.985	0.944	0.971	0.939	0.949	0.722	0.800	0.800	0.979	0.967	0.833	0.667	0.935	0.832	0.969	0.864

TABLE 6
Differences in ranking (mean and standard deviation) between human measurements (H.M.) and between human- and VDHM-obtained results.

k	Library ($N = 104$ el. of interest)				Tourism ($N = 44$ el. of interest)				Accommodation ($N = 80$ el. of interest)				News ($N = 145$ el. of interest)			
[%]	H.M.		VDHM		H.M.		VDHM		H.M.		VDHM		H.M.		VDHM	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
100	1.48	2.60	3.45	4.25	0.95	1.12	1.11	1.24	2.40	3.41	2.90	3.58	1.98	2.96	6.72	10.40
90	1.59	2.72	3.55	4.42	0.98	1.12	1.14	1.25	2.14	2.49	3.13	3.70	2.16	3.07	6.09	9.37
80	1.66	2.87	3.54	4.51	1.20	1.13	0.97	0.98	2.17	2.60	3.25	3.89	2.23	3.22	6.30	9.13
70	1.44	2.86	3.01	4.23	1.30	1.18	1.07	1.01	1.91	2.43	3.55	4.05	2.37	3.40	6.57	9.38
60	1.47	3.05	3.24	4.50	1.23	1.14	1.08	1.09	1.33	1.45	3.13	4.05	2.34	3.14	6.74	9.42
50	1.48	3.30	2.31	3.20	1.27	1.20	1.18	1.14	1.15	1.35	2.60	3.06	1.93	2.77	5.92	8.74
40	0.88	1.94	1.32	1.77	0.94	1.09	1.29	1.21	0.78	0.97	1.19	1.23	1.17	1.78	3.48	7.21
30	0.90	2.20	1.61	1.91	0.62	0.77	0.92	1.12	0.88	1.03	1.25	1.26	1.42	1.97	2.77	2.91
20	0.35	0.59	0.90	0.91	0.75	0.71	1.25	1.28	0.75	0.93	1.38	1.20	1.47	2.11	2.37	2.41
10	0.30	0.67	0.80	0.63	0.50	0.58	0.50	0.58	0.75	1.16	1.25	1.49	0.71	0.73	1.71	1.27

be explained by the fact that the top 30% – 40% represents both elements on which the users focused their attention, both elements in their immediate area. Hence it is easier distinguish them from less viewed portions of the page. Lower values of r and r_S (around $k = 50\%$), instead, were especially found in Library and News scenarios, which correspond to Web pages containing a high amount of information. In those cases, it could be hypothesized that users (slowly) scanned the page looking for the requested information, focusing for few seconds on some elements just to understand whether they were of interest or not. In this case, a human evaluator could potentially recognize users' intentions and assign a lower value to scanned elements, whereas the automatic system has not been proved (yet) to recognize this behavior.

To better understand the impact of the above results it could be useful to consider Table 6, which reports mean value and standard deviation for the difference in ranks between the two evaluators and between manually- and automatically-obtained results. As a matter of example, it can be easily seen that, in the first worst case considered (Tourism), the average difference in ranks was exactly the same, namely 0.5 (on 44 elements), for both H.S. and VDHM. In the other case (Accommodation), mean difference in ranks was 0.75 and 1.25 (i.e., about one position) for H.M. and VDHM, respectively. In general, for the purpose of studying how user's attention is distributed over page elements, a difference of few positions in the ranking could be accepted. Library and News scenarios report the highest differences between manually- and automatically-obtained results, but it should be noted that this value is influenced by the higher number of elements in the page.

In summary, results of the second test phase showed that measurements produced by the proposed method well approximate those obtained by human operators.

6 Conclusions and Future Work

In this paper, the design and development of a framework supporting the collection of data pertaining Web user's activity across heterogeneous devices, and their visualization in an aggregated heat map, is presented. The ultimate goal is to provide Web analysts with a way to study how the attention of multiple users was focused on the various page elements, with the aim to ease the study of the relevance of information displayed, optimize content organization, etc.

Approaches adopted in tools developed so far assume that attention received by a given element can be measured by considering the amount of time that the page region containing the element, i.e., the viewport, was displayed to the user. Unfortunately, modern Web pages that are designed to adapt to device characteristics limit the applicability of heat maps, since elements contained in a given page region might change depending on access configuration. Some tools exist, which draw aggregate heat maps based on navigation data on different devices, but, in some cases, they provide incorrect results. To cope with the above limitation, the devised framework combines information about the viewport with data about the position, size and visibility of each page element in order to produce an element-based, rather than a viewport-based visualization. Such visualization can account for dynamic changes occurring to page aspect and is independent of the access devices used.

An evaluation was carried out to estimate the accuracy in identifying page elements where user's attention was focused onto during navigation (in this case, obtained results were compared with the ones obtained by a state of the art tool), and assess how aggregation capabilities compare with results that could be obtained through manual processing on conventional graphics representations. Experimental results confirmed the effectiveness of the proposed approach.

Future work will be aimed to characterize the overhead introduced by the additional processing required, compared to existing tools, to manage per-element information. Moreover, optimizations possibilities regarding the communication protocol and the algorithm for generating the maps will be investigated. Lastly, experimental tests will be extended to measure framework accuracy and coherency also for pages incorporating dynamic behaviors.

References

[1] M. V. Moreno, F. Terroso-Saenz, A. Gonzalez, M. Valdes-Vela, A. F. Skarmeta, M. A. Zamora-Izquierdo, and V. Chang, "Applicability of Big Data techniques to smart cities deployments," IEEE Transactions on Industrial Informatics, vol. PP, no. 99, pp. 1-1, 2016.

[2] B. Yuan, and B. Falchuck, "Management and visualization of user and network data," China Comm., vol. 10, no. 3, 2013, pp. i-ii.

[3] Q. Z. Sheng, A. V. Vasilakos, Q. Yu, and L. Yao, "Guest editorial: Big Data analytics and the Web," IEEE Transactions on Big Data, vol. 1, no. 4, 2015, pp. 123-124.

[4] Atlas Analytics. Why Data Matters To Your Business. Available online: https://atlasanalytics.io/data-is-the-oil-21st-century/

[5] S. Pengnate, and R. Sarathy, "Visual appeal of websites: The durability of initial impressions," Proc. 46th Hawaii Int. Conf. on System Sciences, 2013, pp. 1530-1605.

[6] V. Navalpakkam, and E. Churchill, "Mouse tracking: Measuring and predicting users' experience of web-based content," in Proc. Human Factors in Comp. Sys., 2012, pp. 2963-2972.

[7] E. Arroyo, T. Selker, and W. Wei, "Usability tool for analysis of web designs using mouse tracks," in Proc. Human Factors in Comp. Sys., 2006, pp. 484-489.

[8] F. Diaz, R. White, G. Busher, and D. Liebling, "Robust models of mouse movement on dynamic web search results pages," in Proc. 22nd Int. Conf. on Inf. and Know.

[9] J. Huang, R. White, and S. Dumais, "No clicks, no problem: Using cursor movements to understand and improve search," in Proc. Human Factors in Comp. Sys., 2011, pp. 1225-1234.

[10] C. Kinney, "United States Patent and Trademark Office," reg. #75263259, September 1, 1993.

[11] F. Lamberti, and G. Paravati, "VDHM: Viewport-DOM based heat maps as a tool for visually aggregating Web users' interaction data from mobile and heterogeneous devices," in Proc. Int. Conf. on Mobile Services, 2015, pp. 33-40.

[12] D. Keim, H. Qu, and K.L. Ma, "Big-Data Visualization," IEEE Computer Graphics and Applications, vol. 33, no. 4, 2013, pp. 20-21.

[13] A. Lex, M. Streit, C. Partl, and D. Schmalstieg, "Comparative analysis of multidimensional, quantitative data," IEEE Transactions on Vis. and Computer Graphics, vol. 16, no. 6, 2010, pp. 1027-1035.

[14] C. Ehmke, and S. Wilson, "Identifying web usability problems from eye-tracking data," in Proc. 21st British HCI Group Annual Conf. on People and Computers, 2007, pp. 119-128.

[15] M. C. Chen, J. R. Anderson, and M. H. Sohn, "What can a mouse cursor tell us more? Correlation of eye/mouse movements on web browsing," in Proc. Human Fact. in Comp. Sys., 2001, pp. 281-282.

[17] Huang, and A. Diriye, "Web user interaction mining from touch-enabled mobile devices," in Proc. 6th Workshop on Human-Computer Interaction and Information Retrieval, 2012, pp. 1-4.

[16] T. Joachims, "Optimizing search engines using clickthrough data," in Proc. 8th Conf. on Know. Disc. and Data Min., 2002, pp. 133-142.

[18] Q. Guo, H. Jin, D. Lagun, S. Yuan, and E. Agichtein, "Mining touch interaction data on mobile devices to predict web search result relevance," in Proc. 36th Int. Conf. on Research and Development in Information Retrieval, 2013, pp. 153-162.

[19] Q. Guo, S. Yuan, and E. Agichtein, "Detecting success in mobile search from interaction," in Proc. 34th Int. Conf. on Research and Development in Information Retrieval, 2011, pp. 1229-1230.

[20] F. Lettner, and C. Holzmann, "Heat maps as a usability tool for multi- touch interaction in mobile applications," in Proc. Int. Conf. on Mobile and Ubiquitous Multimedia, 2012, pp. 1-4.

[21] M. Chae, "Do size and structure matter to mobile users? An empirical study of the effects of screen size, information structure, and task complexity on user activities with standard web phones," Behaviour and Information Technology, vol. 23, 2004, pp. 165-181.

Fabrizio Lamberti is an Associate Professor at Politecnico di Torino, Italy. He received his M.S. and the Ph.D. degrees in computer engineering from Politecnico di Torino in 2000 and 2005. His research interests are in the areas of intelligent systems, HMI, computer graphics, and visualization. He serves as an Associate Editor for IEEE Transactions on Emerging Topics in Computing and for IEEE Consumer Electronics Magazine.

Gianluca Paravati is an Assistant Professor at Politecnico di Torino, Italy, where he received his B.Sc. and M.Sc. degrees in electronic engineering and Ph.D. degree in computer engineering in 2005, 2007, and 2011. His research interests include computer graphics, human-computer interaction, and machine learning.

Valentina Gatteschi is a Postdoctoral Research Assistant at Politecnico di Torino, where she received her B.Sc. and M.Sc. degrees in management engineering and the Ph.D. degree in computer engineering in 2005, 2008 and 2013. Her main research interests are in the fields of semantics, natural language processing, learning technologies and visualization.

Alberto Cannavò received his B.Sc. degree from University of Messina, Italy, in 2013. He received his M.Sc. degree in computer engineering from Politecnico di Torino, Italy, in 2015. Since November 2016 he is pursuing his Ph.D. at Politecnico di Torino, where he is carrying out his research activities in the areas of computer graphics, HMI and HRI.