March 12, 2015

Assisted Browsing via Dynamic Simulated Gravity for Page Elements

Kathleen Michel

Samuel Beder

Assisted Browsing via Dynamic Simulated Gravity for Page Elements

ABSTRACT

A gravity probabilistic system for assisted browsing is disclosed. The system proposes creating a probabilistic model analogous to gravity which attracts and guides the cursor towards clickable UI elements. A notion of mass is used to determine the strength of the attraction, where more important elements have greater mass and attraction. The system determines a relative mass of a UI element on a web page. When the system receives an input related to a cursor movement from a user of the web page, the system moves the cursor according to the relative mass of the UI element.

PROBLEM STATEMENT

Traditional web browsing requires precision in interacting with and selecting user interface (UI) elements on a screen, which is challenging for users who lack fine motor control or sharp vision. Existing accessibility solutions often require purchasing specialized hardware and investing significant time in set-up, which limits their ability to serve a broad range of users and devices. Assisted software for web browsing currently exists in the form of screen readers, voice recognition, assisted/modified mice for inputting commands on a computer and adjusted screen contrast, etc. A model that uses gravity to guide a user towards a UI element is disclosed.

GRAVITY PROBABILISTIC SYSTEM

The systems and techniques described in this disclosure relate to a gravity probabilistic system. The system can be implemented for use in an Internet, an intranet, or another client and server environment. The system can be implemented locally as program instructions on a client device or implemented across a client device and server environment. The client device can be any electronic device such as a mobile device, a smartphone, a tablet, a handheld electronic device, a wearable device, a laptop etc.

Fig. 1 illustrates an example method 100 which describes a system that moves a cursor towards a user interface (UI) element based on a relative mass of the UI element. Method 100 can be performed by the gravity probabilistic system.

A user may be visiting a web page on a web browser. Typically, each of the web pages include a number of UI elements such as input boxes, links, tabs, buttons, etc. The gravity probabilistic system determines a relative mass of a UI element on the web page (block 110). The relative mass of the UI element is a mass of the UI element with respect to mass of other UI elements on the web page. The system determines the relative mass of each of the UI elements by utilizing a probabilistic model created by the system analogous to gravity. A notion of "mass" is used to determine the strength of the attraction for the UI elements, where more important elements have greater mass and attraction. In order to construct the probabilistic model, the system utilizes input from multiple sources. The input sources train the model to strengthen attraction to more important UI features, effectively prioritizing the important elements on a page.

One source of input utilized by the system can include data from multiple users' browsing behavior. More commonly clicked on elements may be given greater mass while occasionally clicked elements may have a smaller mass. Non-elements on a page that are rarely clicked on, for example, text in an article, may be treated as empty space and may be assigned a zero mass. Additionally, the system can take the data from unassisted users as well as gravity-enabled users. The data from gravity-enabled users may have a greater influence as their behavior more closely mirrors that of the target users. The mass for each element may be continuously updated as the user behavior changes.

Additionally, or alternatively, another input source to the model may comprise of data from webmasters, who can explicitly assign gravity strength to UI elements. As well known, a webmaster also known as a web architect, web developer, site author, website administrator, or a website coordinator is a person responsible for maintaining one or many websites. One means of achieving this may be to have webmasters embed accessibility data into their sites via XHTML. XHTML tags could be assigned via a priority scale from 0 to 1.0 (e.g. <accessibility importance= “.5”> <ui element> </accessibility>) to factor in the designer’s intent in emphasizing elements in the gravity model.

Additionally, or alternatively, another input source to the model may include historical browsing habits of the user browsing the web page. The previous interaction with page elements of the user may help prioritizing features frequently used by the user.

The system can use any one of the above input sources alone or use two or more input sources together in combination or all three in combination to create the probabilistic model. The model can be utilized by the system to determine the relative mass of each UI element with respect to each other. The system is aware about the mass assigned to each UI element and can determine the relative mass of each of the UI elements by comparing the mass assigned to each

UI element with another UI element. For example, a button A which is clicked more often will have more mass as compared to a button B which is not clicked as often as button A and hence will have a higher relative mass.

The system receives an input related to a cursor movement from the user (block 120). For example, the user surfing a web page may want to sign up for the website. Considering that a sign up button is situated below the the top right corner of the website, the user may move the cursor towards the button. After receiving the input related to the cursor movement from the user, the system moves the cursor towards a particular UI element based on the relative mass of the UI element (block 130). For example, if the user brings the cursor close to the sign up button and the sign up button has higher relative mass as compared to a input box for entering user's first name situated close to the sign up button, the cursor may be moved towards the sign up button. The system attracts the cursor towards the UI element that has higher mass compared to other UI elements. Thus, when the cursor is moved closer to one of the UI elements, the UI element attracts the cursor toward its center, at which it could be easily clicked by the user. In the above example, there may be other buttons situated in proximity to the sign up button, for example, an input box for entering first name, another input box for entering last name, etc. The system moves the cursor towards the sign up button based on the relative mass of the sign up button with respect to other UI elements.

Additionally, the system can model an area around a button where the cursor is drawn in. For example, as a user approaches an element on a page, the element can light up with a halo effect around the object and the cursor will change shape to show the user that gravity is taking effect. When "locked-in" the cursor can begin to pulsate, showing that the element is currently selectable. The system can also avoid pulling the cursor in when the cursor is far off from the object (drop-off point). Additionally, the system can make the cursor achieve a mild "lock-in" effect when the cursor is very close to element, making it easy to select the element. The system also sets up reasonable escape velocity i.e., appropriate amount of pull to get away from an object.

Additionally, in modeling gravity by the system, the inverse-squared law can serve to define a distance after which the gravitational attraction has no measurable effect, preventing elements pulling the cursor when it is far away. The system uses this concept to set a distance threshold after which that gravity has no effect outside this range.

Fig. 2 illustrates an example implementation of the gravity probabilistic system. The gravity probabilistic system displays an illustrative webpage opened on a web browser on the system. The web page includes multiple UI elements such as website name 210, input box for inputting in username 220, input box for inputting in password 230, login button 240, and text written on the web page 250. The system creates a gravity probabilistic model by retrieving input from multiple sources such as input from multiple users' browsing behaviour, input from the webmasters, and/or input from the historical browsing habits of the user accessing the system. In an example, the system associates most mass with the login button since the login button is pressed most number of times. Now, when the system receives an input related to a cursor movement from the user the system moves the cursor towards the login button 240.

Additionally, or alternatively, the users can vary in terms of their need for assistance in web browsing. Users can choose to enable or disable gravity model. For gravity enabled user, a calibration mode can also be made available to help determine the appropriate amount of assistance required. This calibration can be used in order to set a relative constant applied to the mass of all objects based on the individual users needs. Calibration would likely entail a test in which users try to navigate to an exact point and then stay as close to that point as possible for a set period of time. This calibration would not be necessary to successfully use gravity but would allow users to gain greater customization and accuracy. As users engage with the system, it can further correct itself by tweaking the gravity constant to improve user's accuracy in performing tasks (i.e. success rate) and time to complete tasks against benchmarks (i.e. performance compared to similar users).

Fig. 3 is a block diagram of an environment that shows components of a system for implementing the techniques described in this disclosure. The environment includes client devices 310, servers 330, and network 340. Network 340 connects client devices 310 to servers 330. Client device 310 is an electronic device may be capable of requesting and receiving data/communications over network 340. Example client devices 310 are personal computers (e.g., laptops), mobile communication devices, (e.g. smartphones, tablet computing devices), set-top boxes, game-consoles, embedded systems, and other devices 310' that can send and receive data/communications over network 340. Client device 310 may execute an application, such as a web browser 312 or 314 or a native application 316. Web applications 313 and 315 may be displayed via a web browser 312 or 314. Server 330 may be a web server capable of sending, receiving and storing web pages 332. Web page(s) 332 may be stored on or accessible via server 330. Web page(s) 332 may be associated with web application 313 or 315 and accessed using a web browser, e.g., 312. When accessed, webpage(s) 332 may be transmitted and displayed on a client device, e.g., 310 or 310'. Resources 318 and 318' are resources available to the client device 310 and/or applications thereon, or server(s) 330 and/or web pages(s) accessible therefrom, respectively. Resources 318' may be, for example, memory or storage resources; a text, image, video, audio, JavaScript, CSS, or other file or object; or other relevant resources. Network 340 may be any network or combination of networks that can carry data communication.

The subject matter described in this disclosure can be implemented in software and/or hardware (for example, computers, circuits, or processors). The subject matter can be implemented on a single device or across multiple devices (for example, a client device and a server device). Devices implementing the subject matter can be connected through a wired and/or wireless network. Such devices can receive inputs from a user (for example, from a mouse, keyboard, or touchscreen) and produce an output to a user (for example, through a display). Specific examples disclosed are provided for illustrative purposes and do not limit the scope of the disclosure.

DRAWINGS

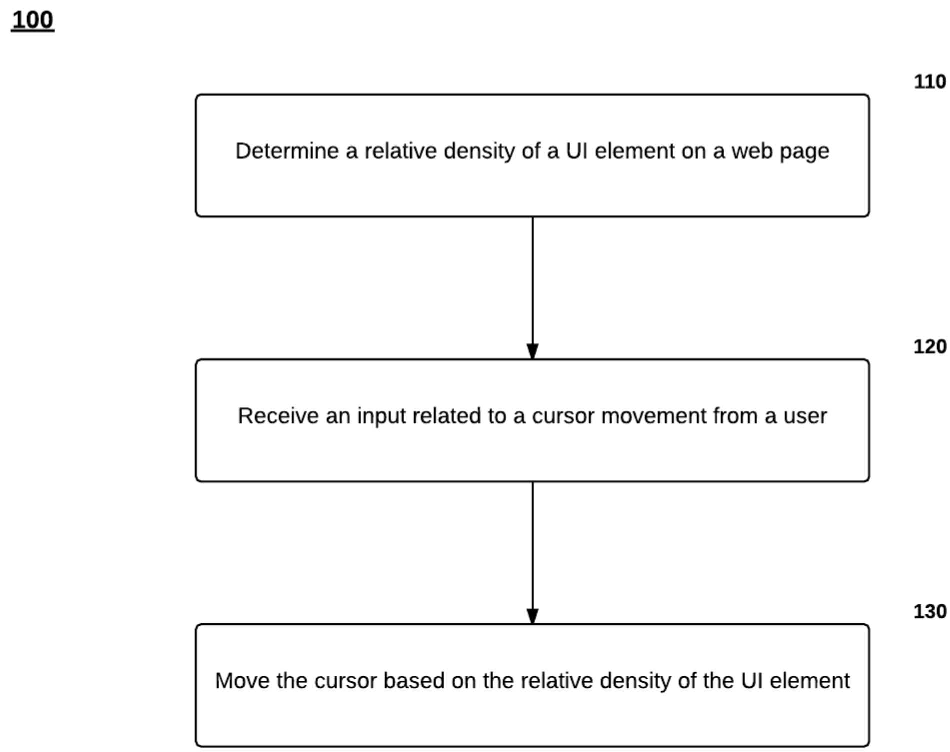
100
110
Determine a relative density of a UI element on a web page
120
Receive an input related to a cursor movement from a user
130
Move the cursor based on the relative density of the UI element

FIG. 1

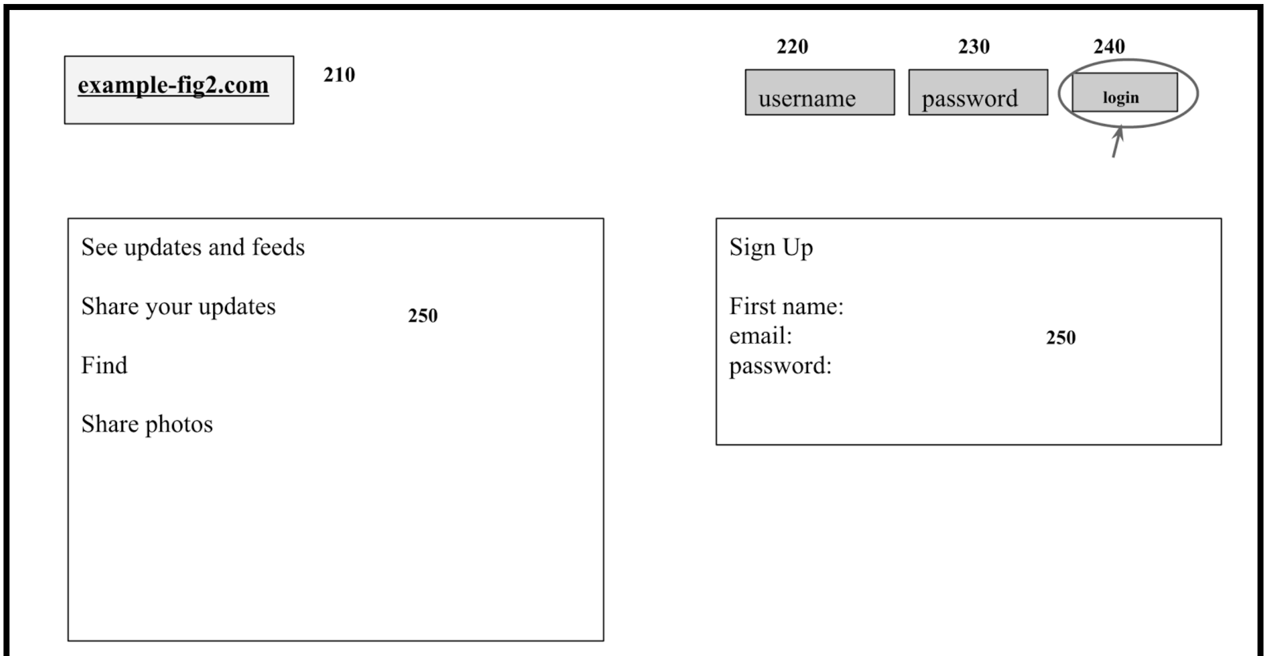
example-fig2.com
210
220
230
240
username
password
login
See updates and feeds
Share your updates
250
Find
Share photos
Sign Up
First name:
email:
password:
250

FIG. 2

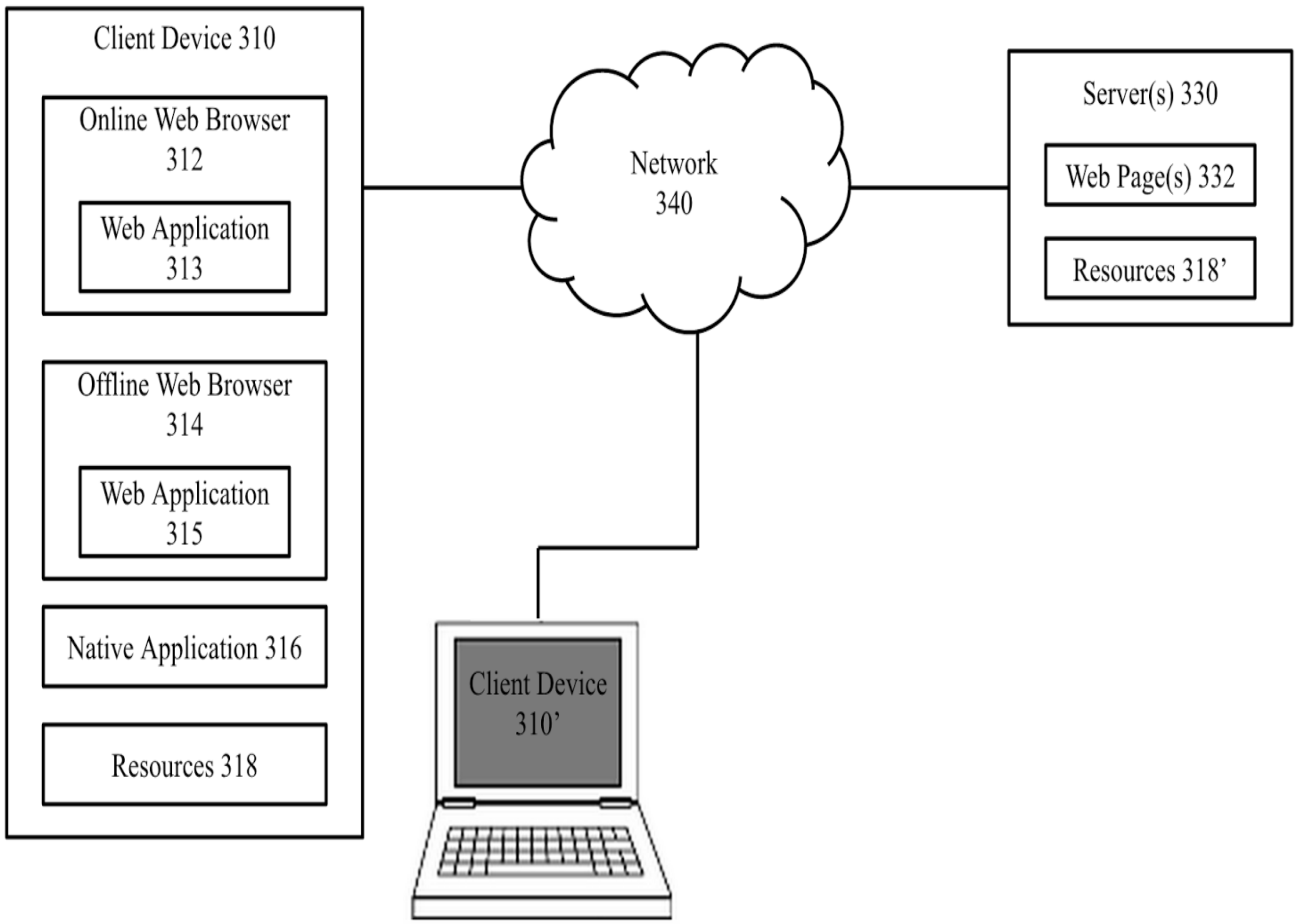
Client Device 310
Online Web Browser 312
Web Application 313
Offline Web Browser 314
Web Application 315
Native Application 316
Resources 318
Network 340
Server(s) 330
Web Page(s) 332
Resources 318'
Client Device 310'

Fig. 3