

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
29 October 2009 (29.10.2009)

PCT

(10) International Publication Number  
**WO 2009/132148 A2**

(51) International Patent Classification:

G06F 3/03 (2006.01) G06F 17/00 (2006.01)  
G06F 21/00 (2006.01)

(21) International Application Number:

PCT/US2009/041462

(22) International Filing Date:

22 April 2009 (22.04.2009)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/047,071 22 April 2008 (22.04.2008) US

(71) Applicant (for all designated States except US): **THE 41ST PARAMETER, INC.** [US/US]; 1785 1 No. 85th Street, Ste. 250, Scottsdale, AZ 85255 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **EISEN, Ori** [US/US]; 6214 E Hillery Drive, Scottsdale, AZ 85254 (US). **PANDICH, Stephen, George** [US/US]; 20100 North 78th Place #2140, Scottsdale, AZ 85255 (US).

(74) Agents: **ENG, U.P. Peter** et al; Wilson Sonsini Goodrich & Rosati, 650 Page Mill Road, Palo Alto, CA 94304-1050 (US).

(81) Designated States (unless otherwise indicated, for every

kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every

kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(gf))

(54) Title: SYSTEMS AND METHODS FOR SECURITY MANAGEMENT BASED ON CURSOR EVENTS

(57) Abstract: Systems and methods for security management based on cursor events. Click coordinates on a webpage of user's activity are logged in order to detect multiple or repeated transactions that bare the same exact coordinates. Such multiple or repeated activity may be helpful in detecting scripted attacks or bot net activity in a variety of environments and transactions. Further, the invention may also capture the coordinates of the location of submit buttons, and similar buttons, on a webpage to analyze whether a click action or other pointer action occurred within the button's parameters.



WO 2009/132148 A2

SYSTEMS AND METHODS FOR SECURITY MANAGEMENT BASED ON CURSOR  
EVENTS

## CROSS-REFERENCE

5

[0001] This application claims the benefit of U.S. Provisional Application No. 61/047,071 filed April 22, 2008, which application is incorporated herein by reference.

## BACKGROUND

10

[0002] Automatic scripts present a number of problems faced by Internet users today. In one scenario, Internet users may receive SPAM or junk e-mail that has relatively little worth to most e-mail users but that may clutter e-mail inboxes, expend a great deal of bandwidth on the Internet, and result in slower response times and decreased efficiency. One of the main causes of SPAM are computers that use scripts to impersonate users. During the registration process for an e-mail address, a computer can use a script to generate multiple e-mail addresses that do not even have a corresponding human user. These multiple e-mail addresses can then be used by the computer to send unwanted e-mail messages to other Internet users.

15

[0003] Computers can be used to run scripts that send out multiple e-mail messages from a user generated e-mail address or from a script generated e-mail address to the same set of recipients. A computer user would take a great deal of time to send out a large number of e-mail messages to a set of recipients. On the other hand, a computer running a script can transmit e-mail messages at the speed that it would take thousands of computer users to perform the same task in the same amount of time.

20

[0004] Another area of services in which scripts pose a large problem is Internet voting services and Internet contests. The Internet has many web pages in which users can participate in a poll. For instance, users can vote for their favorite musician, artist, athlete, etc. The user can always return to the web page and vote another time, but the chances are small that most users will bother trying to vote again. However, a computer can use a script to vote a large number of times in a very small amount of time. The result is that the voting results are skewed by a large amount. Further, a user that enters a contest will almost never win if a script places thousands of entries into the same contest.

30

[0005] Bot nets may be a collection of software robots, or bots, which run autonomously and automatically on groups of computers controlled remotely. The term "bot net" is also used to refer to any group of bots, a collection of compromised computers running programs, usually referred to as worms, Trojan horses, or backdoors, under a common command and control infrastructure. The bot nets may be controlled remotely, can run hidden, and can comprise the security of systems. Some bot nets may scan environments and propagate themselves using vulnerabilities and weak passwords.

[0006] Bot nets have been exploited for various purposes, including denial-of-service attacks, creation or misuse of SMTP mail relays for SPAM, click fraud, spamdexing and the theft of application serial numbers, login IDs, and financial information such as credit card numbers. For example, a bot net may be created and used to send e-mail SPAM. First, a bot net operator may send out viruses or worms, infecting an ordinary user's computer. Then, the "bot" on the infected computer may log into a particular server, which becomes the command-and-control server. Next, a spammer may purchase access to the bot net from the bot net operator, and the spammer may send instructions via the server to the infected computers, causing them to send out spam messages to mail servers.

[0007] Online banking services, online shopping applications, and other systems which require the entry of passwords or other sensitive information in a public environment are all susceptible to compromise. What are needed are systems and methods for security management that are user-friendly, not readily susceptible to observation during data entry, and resistant to interception and computational deciphering techniques.

### SUMMARY OF INVENTION

[0008] The invention provides systems and methods for security management based on cursor events. Various aspects of the invention described herein may be applied to any of the particular applications set forth below. The invention may be applied as a standalone security system or as a component of an integrated solution for security management. The invention can be optionally integrated into existing business and security management processes seamlessly. It shall be understood that different aspects of the invention can be appreciated individually, collectively or in combination with each other.

[0009] It shall be understood that a variety of pointing devices are contemplated for use in the invention, including a mouse, touch-pad, trac-point, nub, trackball, etc. or any device that moves a cursor or pointer on a visual display. The invention shall not be limited by the examples described herein.

[0010] One aspect of the invention describes a method for security management. The invention may provide for security management by receiving and logging coordinate data for a mouse or other pointing device action by a user. The coordinate data may include X, Y coordinates on a display screen or within a web browser of the location of the pointer or cursor corresponding to the user's mouse or other pointing device. The system may then analyze the logged coordinate data, detect patterns of repetitive coordinate data, and then may proceed to block network attacks based on the patterns of repetitive coordinate data.

[0011] Another aspect of the invention provides a system for security management with a validation module for logging the coordinates of the pointer or cursor and detecting patterns of repetitive coordinate data. The system may obtain the parameters of the coordinates corresponding to a click of a pointing device on a web page and then log and analyze the coordinates. Because many transactions may bare the exact same coordinates, it may help detect scripted attacks or bot net activity. In addition, the system may obtain the parameters of the coordinates of the location of the button that the click action acted upon, to analyze whether the click action actually occurred within the coordinates of the button.

[0012] Unless specifically stated otherwise, as apparent from the following discussions, it is appreciated that throughout the specification, discussions utilizing terms such as "processing," "computing," "calculating," "determining," or the like, may refer in whole or in part to the action and/or processes of a processor, computer or computing system, or similar electronic computing device, that manipulate and/or transform data represented as physical, such as electronic, quantities within the system's registers and/or memories into other data similarly represented as physical quantities within the system's memories, registers or other such information storage, transmission or display devices. It will also be appreciated by persons skilled in the art that the term "users" referred to herein can be individuals as well as corporations and other legal entities. Furthermore, the processes presented herein are not inherently related to any particular computer, processing device, article or other apparatus. An example of a structure for a variety of these systems will appear from the description below. In addition, embodiments of the present invention are not described with reference to any particular processor, programming language, machine code, etc. It will be appreciated that a variety of programming languages, machine codes, etc. may be used to implement the teachings of the invention as described herein.

[0013] Other goals and advantages of the invention will be further appreciated and understood when considered in conjunction with the following description and accompanying drawings. While the following description may contain specific details describing particular

embodiments of the invention, this should not be construed as limitations to the scope of the invention but rather as an exemplification of preferable embodiments. For each aspect of the invention, many variations are possible as suggested herein that are known to those of ordinary skill in the art. A variety of changes and modifications can be made within the scope of the invention without departing from the spirit thereof.

### INCORPORATION BY REFERENCE

[0014] All publications and patent applications mentioned in this specification are herein incorporated by reference to the same extent as if each individual publication or patent application was specifically and individually indicated to be incorporated by reference.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0015] Some of the features of the invention are described as set forth in the following figures and description. A better understanding of the features and advantages of the invention will be obtained by reference to the following detailed description that sets forth illustrative embodiments provided in accordance with the invention.

[0016] FIG. 1 describes an example of a webpage from which coordinate data can be obtained and analyzed for a click action or button.

[0017] FIG. 2 shows an example of a system that may receive and analyze coordinate data from click actions from human users or scripts.

[0018] FIG. 3 shows an example of cursor events that may occur if a human user were clicking on a button on a webpage.

[0019] FIG. 4 shows an example of cursor events that may occur if a script were interacting with a webpage.

[0020] FIG. 5 is a flow diagram showing a method implemented according to the present invention.

[0021] FIG. 6 illustrates one embodiment of analyzing coordinates of one click action with stored coordinates of previous transactions.

[0022] FIG. 7 illustrates an example of a set of webpages from which coordinate data can be obtained and analyzed for a click action or button.

### DETAILED DESCRIPTION OF INVENTION

[0023] In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the invention. However it will be understood by those of ordinary skill in the art that the invention may be practiced without these specific details. In other

instances, well-known methods, procedures, components and circuits have not been described in detail so as not to obscure the invention. Various modifications to the described embodiments will be apparent to those with skill in the art, and the general principles defined herein may be applied to other embodiments. The invention is not intended to be limited to the particular  
5 embodiments shown and described.

[0024] Referring to FIG. 1, on a webpage 100, there may be a variety of buttons that a user may click on with a mouse or other pointing device 104. For example, a user may click on a submit button, an upload button, a verify button, a login button, or a variety of other buttons. It will be appreciated that a variety of other types of buttons or click actions are contemplated by  
10 the invention. It shall be understood that a variety of pointing devices are contemplated for use in the invention, including a mouse, touch-pad, track-point, nub, trackball, etc. or any device that moves a cursor or pointer on a visual display. It shall also be understood and appreciated that a "click action" or "click" may refer to any action by any pointing device, including a tap, selection, etc., and the invention will not be limited to the examples described herein.

[0025] In one embodiment of the invention, when a user clicks on the webpage 100 with a pointing device 104, the X coordinate 106 and the Y coordinate 108 of the pointer action or cursor event may be logged. If the user is a human user and not a computer running a script, then the X and Y coordinates 106, 108 would most likely vary slightly with each transaction. If, however, a scripted attack or bot net activity provided the X and Y coordinates 106, 108, then it  
15 could be the case that many transactions would have the same exact X and Y coordinates 106, 108. In such a case where many transactions bare the same exact coordinates, the scripted attacks or bot net activity could be detected.

[0026] In another embodiment of the invention, the X and Y coordinates of the button 102 are also captured. The X and Y coordinates of the button 102 would be a range of coordinates  
25 that fall within the location of the button 102. When a user clicks on the webpage 100 with a mouse or other pointer device 104, the X coordinate 106 and the Y coordinate 108 of the click action or cursor (pointer) event may be logged. If a user is a human user and not a computer running a script, then the X and Y coordinates 106, 108 would be within the range of coordinates that fall within the location of the button 102. The system could then verify that the click event  
30 actually happened on the button 102. If, however, a scripted attack or bot net activity provided the X and Y coordinates 106, 108, then it could be the case that the X and Y coordinates 106, 108 provided by the scripted attack or bot net activity would fall outside of the coordinates within the location of the button 102. Thus, the system could verify that the click event did not happen on the button 102.

[0027] For example, in an online banking environment, if a user is logging into his or her account, the user may type in his user ID and password, and then click a submit or login button 104. The user's click action may have X coordinate 106 and Y coordinate 108, and those coordinates X, Y may be logged. In one embodiment of the invention, those coordinates X, Y corresponding to a login event, may be stored on a database. The system could verify whether the X and Y coordinates of the click action or cursor event fall within the coordinates of the location of the button 102. Further, the system could compare the X and Y coordinates of the click action 106, 108 to other X, Y coordinates for the transaction that were previously logged or stored. If the X and Y coordinates of the click action 106, 108 are the same exact coordinates of multiple previous transactions, and there is a pattern of repetitive coordinate data, then the system will be able to detect the possibility of a scripted attack or bot net activity.

[0028] Referring to FIG. 2, on any given webpage a user 202 may submit, update, or access information over the Internet 200 by clicking on a button (e.g., a submit button, a login button, an upload button, etc.). When a user clicks on a button, the X and Y coordinates 212 of the cursor event may be sent to an application 222 on a server 220. The application 222 or a validation module 224 may verify whether the X and Y coordinates 212 of the click action fall within the coordinates of the location of the button. Further, the application 222 or a validation module 224 may store the X and Y coordinates 212 of the cursor event on a database 226. The validation module 224 may compare the X and Y coordinates 212 with the X and Y coordinates stored on the database 226. If the validation module 224 detects that many transactions bare the same exact coordinates, then it may alert other applications as to suspected scripted attacks or bot net activity.

[0029] In an embodiment of the invention, if a computer running a script 204 is accessing a webpage on the Internet 200, then it may send X and Y coordinates 214 to an application 222 on a server 220. The application 222 or a validation module 224 may verify whether the X and Y coordinates 214 fall within the coordinates of the location of the button that the X and Y coordinates 214 should correspond to. Thus, the application 222 or a validation module 224 may check to see whether the cursor event actually happened on the button on the webpage or not. Further, the application 222 or a validation module 224 may store the X and Y coordinates 214 of the cursor event on a database 226. The validation module 224 may compare the X and Y coordinates 214 with the X and Y coordinates stored on the database 226. Because the computer is running a script 204, it may be the case that many transactions bare the same exact X and Y coordinates. The validation module 224 may discover that the X and Y coordinates 214 are the

exact same as many of the coordinates stored on the database, and thus can detect and alert other applications as to the suspected scripted attacks or bot net activity.

[0030] As shown in FIG. 3, a human user 202 may, in repeated transactions on a webpage, click on a button 102 with a cursor 104. When a human user 202 clicks on a button 102, the resulting X and Y coordinates 212 associated with the cursor events may vary slightly between clicks or cursor events. It is very unlikely that a human user 202 will click on the exact same X and Y coordinates 212 within button 102 repeatedly. In contrast, as shown in FIG. 4, a script 204 may, in repeated transactions on a webpage, simulate a user that would click on a button 102 with a cursor 104. In contrast to the human user who may produce varying X and Y coordinates 212, the automated script 204 may have repeated X and Y coordinates 214. Thus, if the system stores the X and Y coordinates 214 on a database or in memory, then the X and Y coordinates 214 of the automated script 204 will be identical and the same as previous X and Y coordinates stored on a database or in memory. Thus, a validation module may discover that the X and Y coordinates 214 are the exact same as many of the coordinates on the database. The validation module is then able to detect and alert other applications as to the suspected scripted attacks or bot net activity.

[0031] Further, in an alternative embodiment of the invention, the validation module may also be able to detect patterns of human action. For example, the system may be able to detect clicks that occur in a similar area or zone to help with user authentication. In one embodiment of the invention, the system may be able to detect a human user returning, by observing that the human users generally click in one area on a page. Thus, the individual click events may be compared to a "click heat," which observes in the aggregate where users click on a page.

[0032] In addition, other mouse metrics may be observed, logged and analyzed. Such components include mouse metrics such as accuracy, bias, click duration, confirmation dependency, convexity, double-click rhythm, mouse-down travel/drag, over-click, overshoot and braking, speed and acceleration, and tremor, jerking or wobbling. Such metrics are described in detail in U.S. Patent Publication No. 2005/0008148 (Jacobson) published on January 13, 2005, which is hereby incorporated by reference in its entirety. Thus, such metrics may be used to authenticate a user of a mouse.

[0033] FIG. 5 is a flow diagram showing a method implemented according to the present invention. In step 504, a webpage is loaded in a browser, and has at least one button. The button can be a submit button, an upload button, a find button, a login button, or any kind of button that is on a webpage. It shall be understood that the invention is not limited to any specific type of button, but rather contemplates all buttons that may be on a webpage. In step 506, the location

of the button is stored. In one embodiment, the button's location information may be stored on a database or in memory. The X and Y coordinates of the button may be captured by a range of coordinates that fall within the location of the button. In step 508, when a user clicks on a button within a webpage, the X and Y coordinates of the click action or cursor (pointer) event may be received and stored.

**[0034]** In step 510, a comparison of the button's location information and the X and Y coordinates of the click action or cursor (pointer) event can be made. In one embodiment, the method could verify whether the X and Y coordinates of the click action fall within the coordinates of the location of the button. In step 512, a comparison of the X and Y coordinates of the click action and the X and Y coordinates of previous click actions for previous transactions can be made. The method could compare the X and Y coordinates of the click action of other X and Y coordinates for the transaction that were previously logged or stored. If the X and Y coordinates of the click action are the same exact coordinates as those of multiple previous transactions, then a pattern of repetitive coordinate data can be detected. The method, therefore, is able to detect the possibility of a scripted attack or bot net activity.

**[0035]** FIG. 6 illustrates the coordinates of a click action on a button 600, where coordinates X 612 and Y 614 can be compared to coordinates of previous transactions 622 that are stored on a database 620. The database 620 can be used in other embodiments of the invention, such as shows and described in FIG. 2. Alternatively, the coordinates may be logged or stored in memory. One way that this comparison can be done is illustrated by equation 630. The sum of differences between each previous X and Y coordinate stored on database 620 and the X coordinate 612 and Y coordinate 614 of the current click action must exceed a minimum threshold 632. Thus, the sum of the differences between each of coordinates  $(X_1, Y_1)$ ,  $(X_2, Y_2)$ ,  $(X_3, Y_3) \dots (X_N, Y_N)$  622 stored on database 620 and current coordinates of the click action (X, Y) must meet a minimum threshold 632.  $|X_1 - X| + |X_2 - X| + |X_3 - X| + \dots + |X_N - X| + |Y_1 - Y| + |Y_2 - Y| + |Y_3 - Y| + \dots + |Y_N - Y| \geq \text{minimum threshold } 632$ . This minimum threshold 632 may be a fixed value, or alternatively, may vary based on the number of previous coordinates 622 stored on database 620 for the particular transaction. For example, if there are only two sets of coordinates 622, then the minimum threshold 632 may be lower or higher than if there are a thousand sets of coordinates 622.

**[0036]** Referring to FIG. 7, the invention may also apply to a series of click actions or cursor (pointer) events. For example, if a login process requires three separate steps in which a user must click on three separate buttons, then the invention would apply across those three steps and three different webpages 700, 702, and 704. To illustrate, on webpage 700, a user may click a

button 720 which will result in webpage 702 being loaded. On webpage 702, a user may click a button 722, which results in webpage 704 being loaded. Then, a user may click on button 724, and so on. Instead of only logging or storing one set X and Y coordinates, the invention may log and store both the individual  $X_i$  and  $Y_i$  coordinates associated with button 720 and also a set of coordinates  $(X_i, Y_i)$ ,  $(X_2, Y_2)$ , and  $(X_3, Y_3)$  associated with the transaction. The system may verify that each click event actually happened on each button. Further, the system could compare the set of coordinates  $(X_1, Y_1)$ ,  $(X_2, Y_2)$ , and  $(X_3, Y_3)$  associated with the transaction with previous sets of coordinates stored on a database or in memory. If the set of coordinates are the same exact set of coordinates of multiple previous transactions, and there is a pattern of repetitive coordinate data, then the system will be able to detect the possibility of a scripted attack or bot net activity. When scripted attacks or bot net activity is detected, then the suspect activity may be blocked by the user interface, or an administrator or website administrator or operator may take appropriate action regarding the detection of automated activity.

[0037] In another embodiment, the system according to the present invention may include a sensor (or microsensors) which captures or gathers data relating to the movement of a mouse. A local memory may store the raw data, and software components that are accessible by the client server may communicate the data to a server. On the server side, a comparison may be made of the raw data against the previously logged data in a database, and a validation module may determine a value indicating the likelihood or probability of the user being the authorized person or the likelihood or probability of the user being a human user.

[0038] In addition to detecting scripted or bot net activity for security management, methods and systems described herein may be used to facilitate authentication in a variety of contexts, including but not limited to: customers making an online purchase or any online transaction, release of medical information online, an internet voting system, entering online sweepstakes, purchasing tickets online for an event, etc.

[0039] All concepts of the invention may be incorporated or integrated with other systems and methods of facilitating authentication of users of mouse devices, including but not limited to those described in U.S. Patent Publication No. 2005/0008148 (Jacobson) published on January 13, 2005, which is hereby incorporated by reference in its entirety.

[0040] It should be understood from the foregoing that, while particular implementations have been illustrated and described, various modifications can be made thereto and are contemplated herein. It is also not intended that the invention be limited by the specific examples provided within the specification. While the invention has been described with reference to the aforementioned specification, the descriptions and illustrations of the preferable

embodiments herein are not meant to be construed in a limiting sense. Furthermore, it shall be understood that all aspects of the invention are not limited to the specific depictions, configurations or relative proportions set forth herein which depend upon a variety of conditions and variables. Various modifications in form and detail of the embodiments of the invention will  
5 be apparent to a person skilled in the art. It is therefore contemplated that the invention shall also cover any such modifications, variations and equivalents.

WHAT IS CLAIMED IS:

1. A computer readable medium containing program instructions for detecting automated computer activity within a user interface, comprising:

- 5 selecting a mouse action by a user to be analyzed;
- receiving and logging X and Y coordinates for the mouse action by the user;
- analyzing the logged X and Y coordinates; and
- detecting patterns of repetitive X and Y coordinates.

2. The computer readable medium as in claim 1, wherein the mouse action by a user  
10 to be analyzed is a click action on a button within the user interface.

3. The computer readable medium as in claim 1, further comprising the step of blocking network attacks based on the patterns of repetitive X and Y coordinates.

4. The computer readable medium as in claim 1, further comprising blocking network attacks based on a comparison of logged X and Y coordinates corresponding to a  
15 current click action to previously stored X and Y coordinates corresponding to a history of click actions.

5. A computer system having a computer readable medium that stores a computer program with machine readable instructions for carrying out the method for detecting automated computer activity within a user interface comprising:

20 computer code that enables the receipt of coordinate data including at least one set of coordinates of a user's mouse actions within the user interface; and

computer code that instructs a computer to have a validation module for logging the coordinates and detecting patterns of repetitive coordinate data.

6. The system as in claim 5, wherein the mouse action by a user is a click action on a  
25 button within the user interface.

7. The system as in claim 5, wherein the validation module blocks network attacks based on the patterns of repetitive coordinate data.

8. The system as in claim 5, wherein the validation module blocks network attacks based on a comparison of logged coordinate data corresponding to a current click action to  
5 previously stored coordinate data corresponding to a history of click actions.

9. A method for security management, comprising the steps of:

receiving a plurality of coordinate data for pointer activity selected within a button on a visual display within a user interface;

analyzing the plurality of coordinate data to detect substantially similar coordinate data  
10 corresponding to the visual display.

10. The method for security management as in claim 9, further comprising the step of blocking network attacks based on the substantially similar coordinate data.

11. The method for security management as in claim 9, wherein the coordinate data for pointer activity consists of an X coordinate and a Y coordinate.

12. The method for security management as in claim 9, wherein the step of analyzing  
15 the plurality of coordinate data to detect substantially similar coordinate data further comprises calculating the sum of the differences between each of the coordinates.

13. The method for security management as in claim 12, wherein the sum of the differences is compared to a minimum threshold.

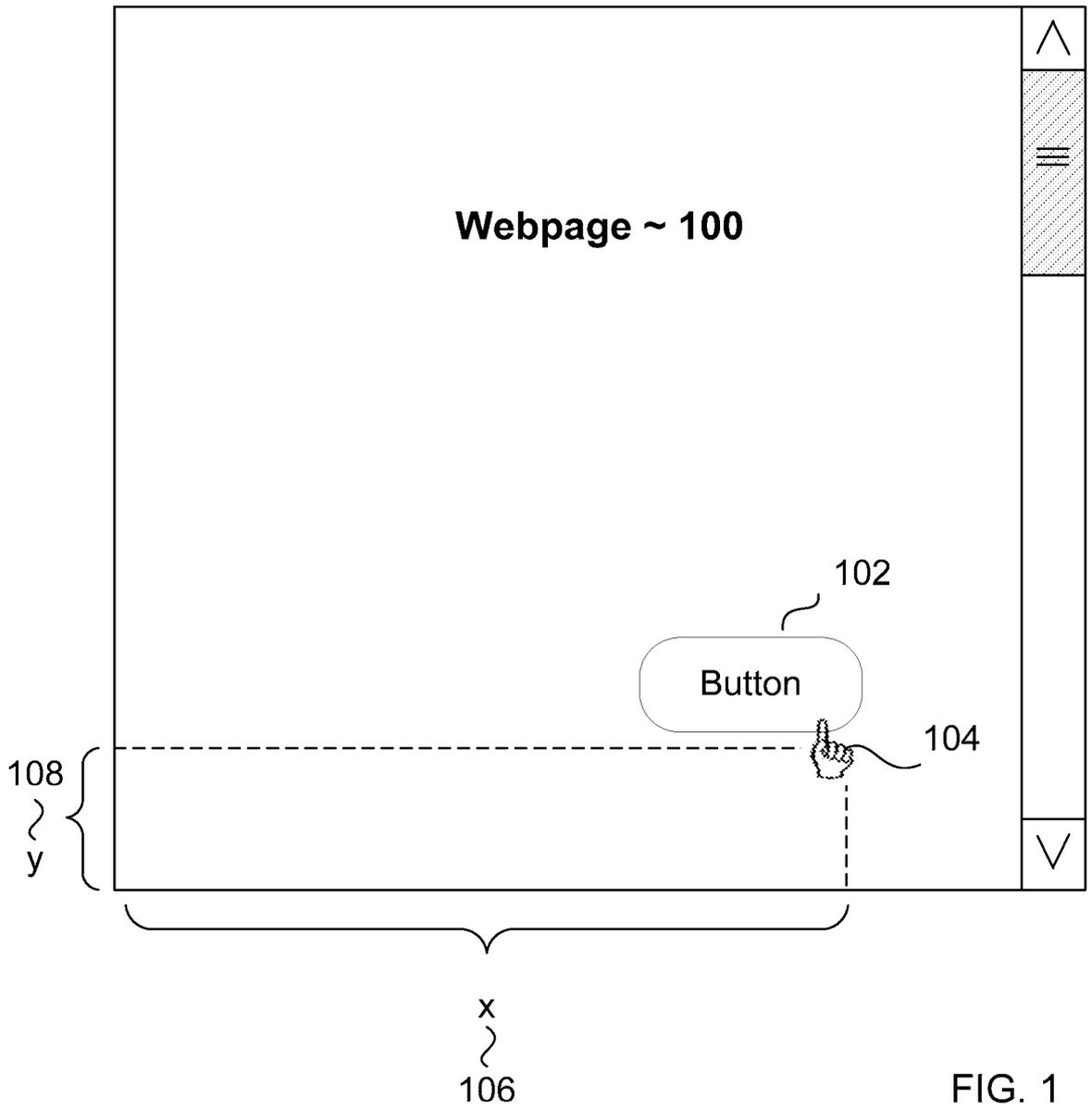
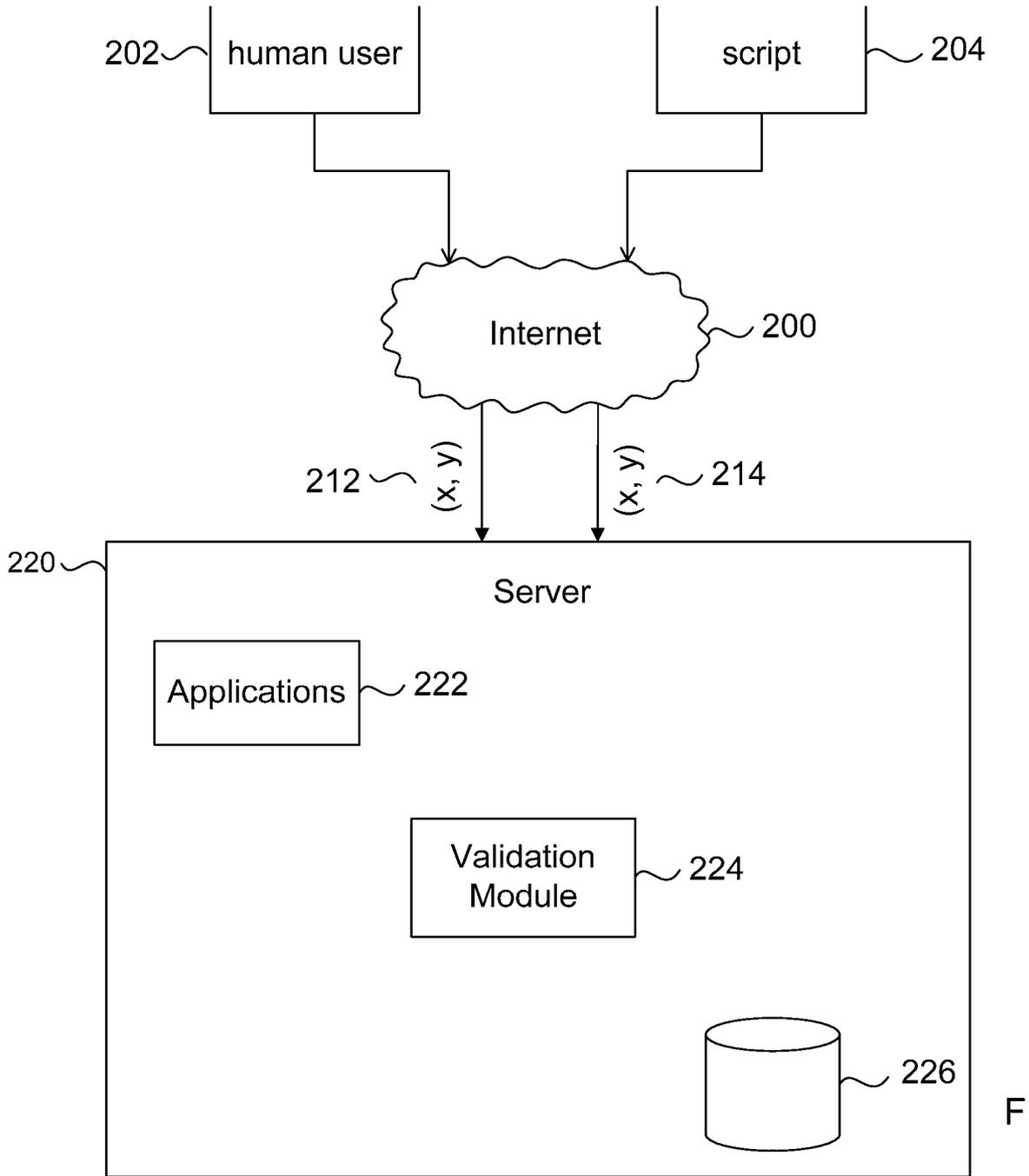


FIG. 1



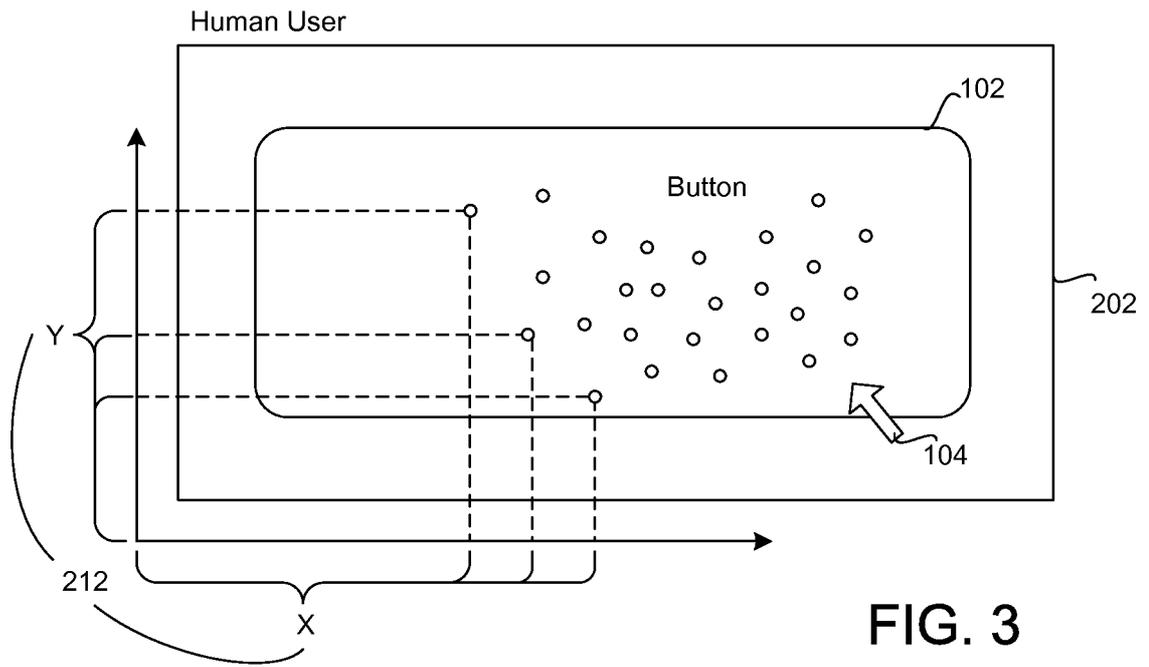


FIG. 3

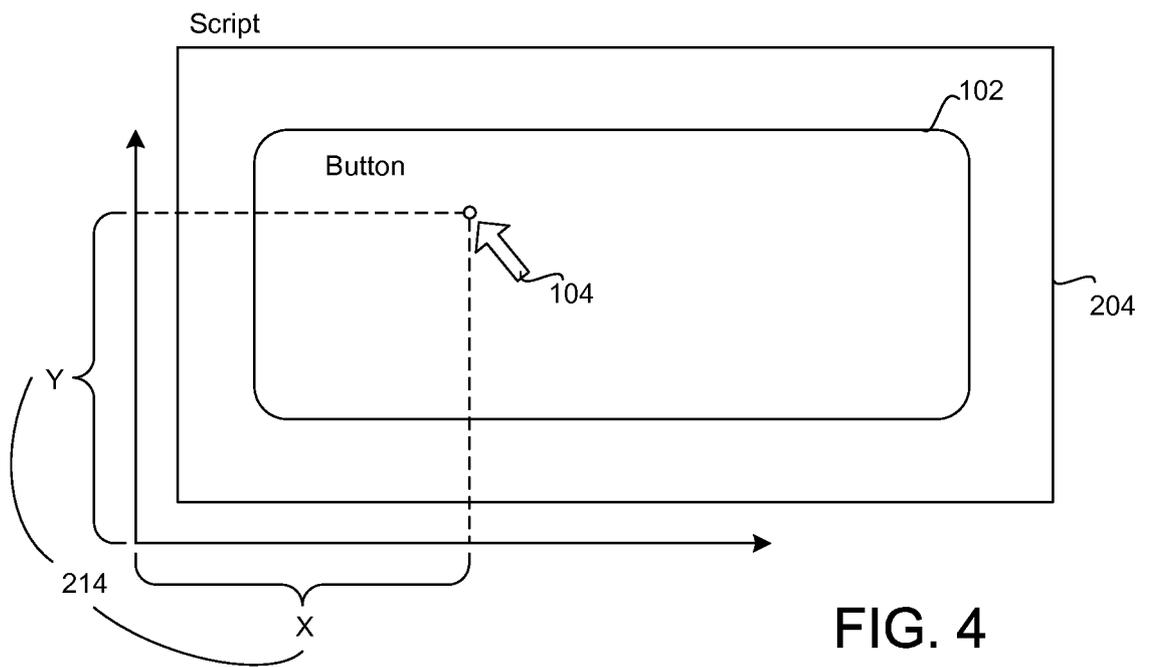


FIG. 4

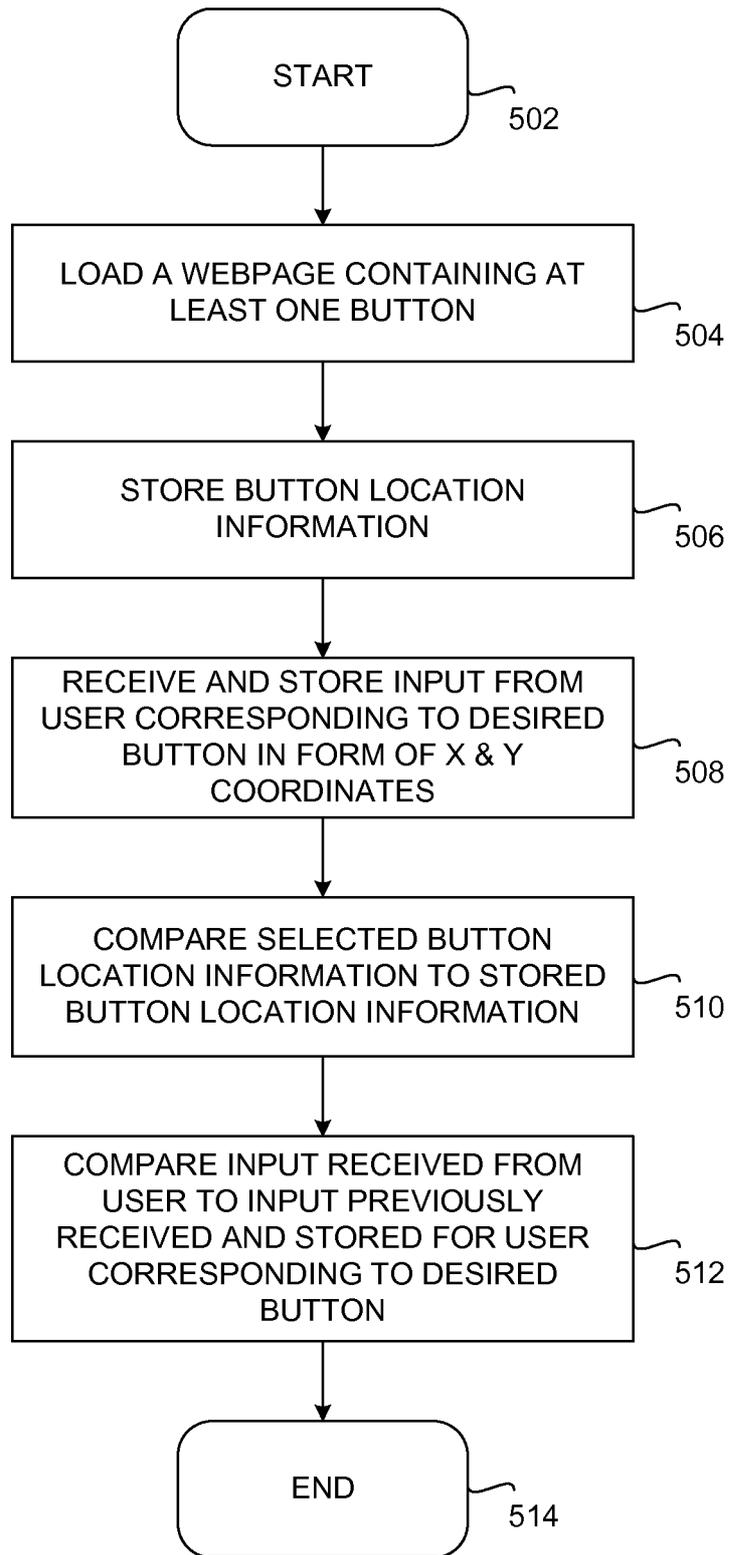


FIG. 5

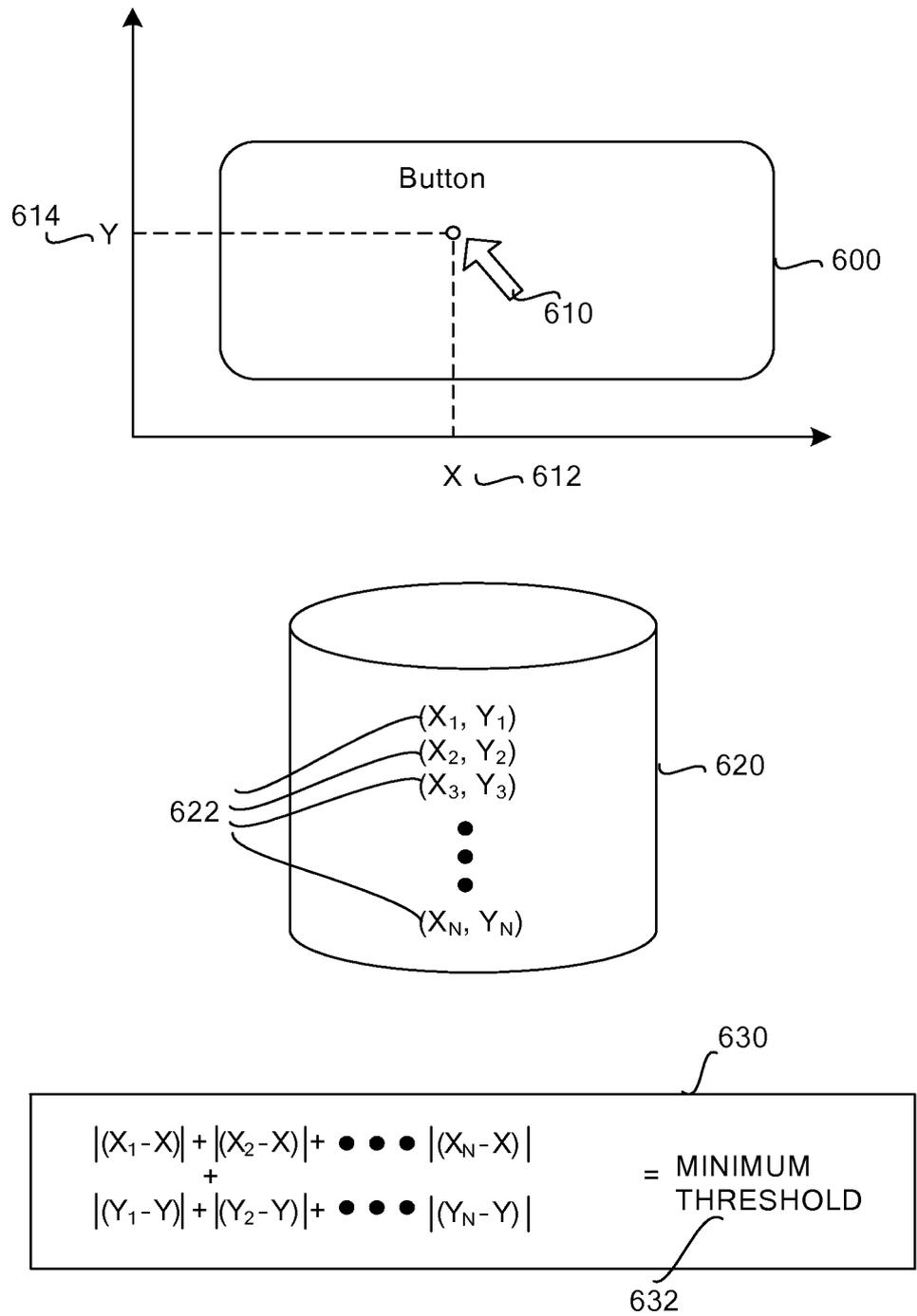


FIG. 6

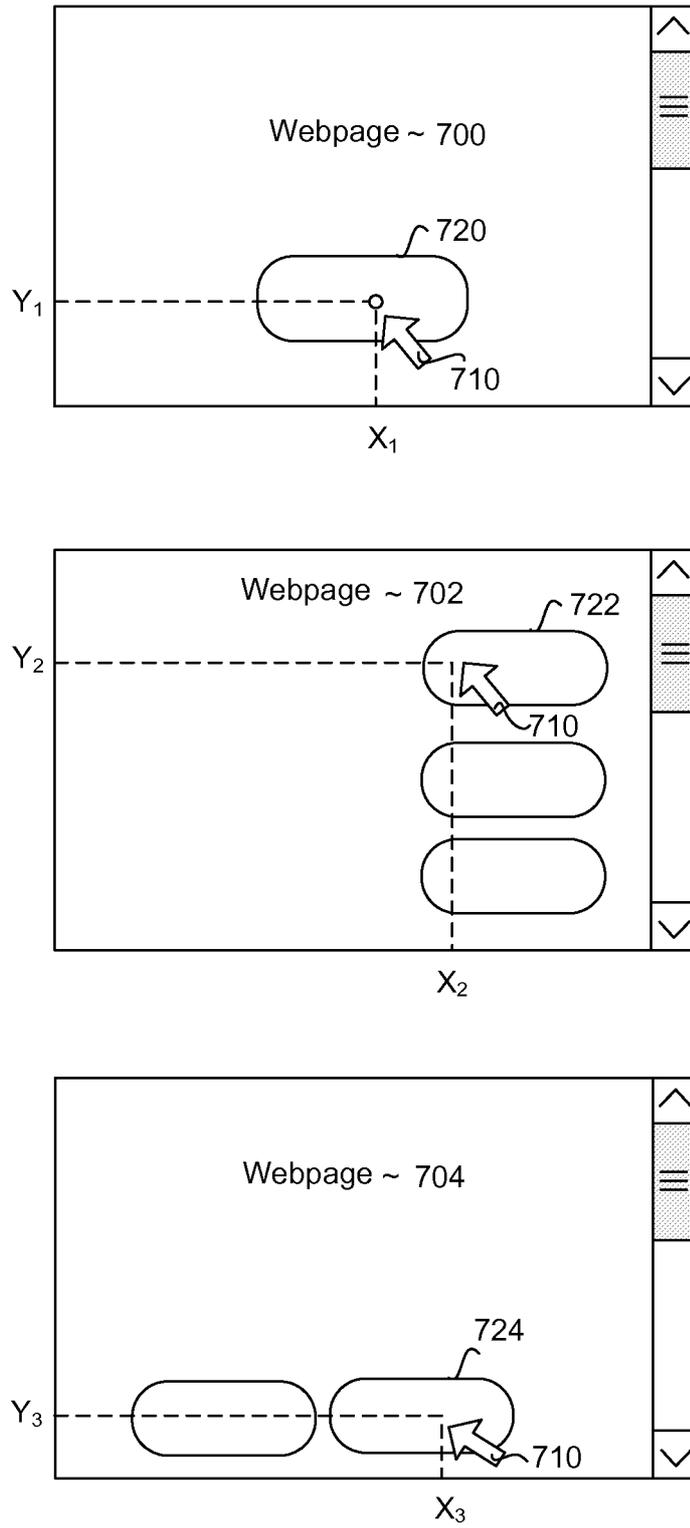


FIG. 7