



US 20100010971A1

(19) **United States**  
(12) **Patent Application Publication**  
**Eyal et al.**

(10) **Pub. No.: US 2010/0010971 A1**  
(43) **Pub. Date: Jan. 14, 2010**

(54) **METHOD AND SYSTEM FOR VISUAL NETWORK SEARCHING**

**Related U.S. Application Data**

(75) Inventors: **Aviv Eyal**, New York, NY (US);  
**Shirley Shor**, New York, NY (US)

(63) Continuation of application No. 11/246,740, filed on Oct. 7, 2005, Continuation of application No. 09/843,286, filed on Apr. 25, 2001, now Pat. No. 7,010,537.

Correspondence Address:  
**BLACK LOWE & GRAHAM, PLLC**  
701 FIFTH AVENUE, SUITE 4800  
SEATTLE, WA 98104 (US)

**Publication Classification**

(51) **Int. Cl.**  
**G06F 7/10** (2006.01)  
**G06F 17/30** (2006.01)  
(52) **U.S. Cl.** ..... **707/3; 707/E17.108**

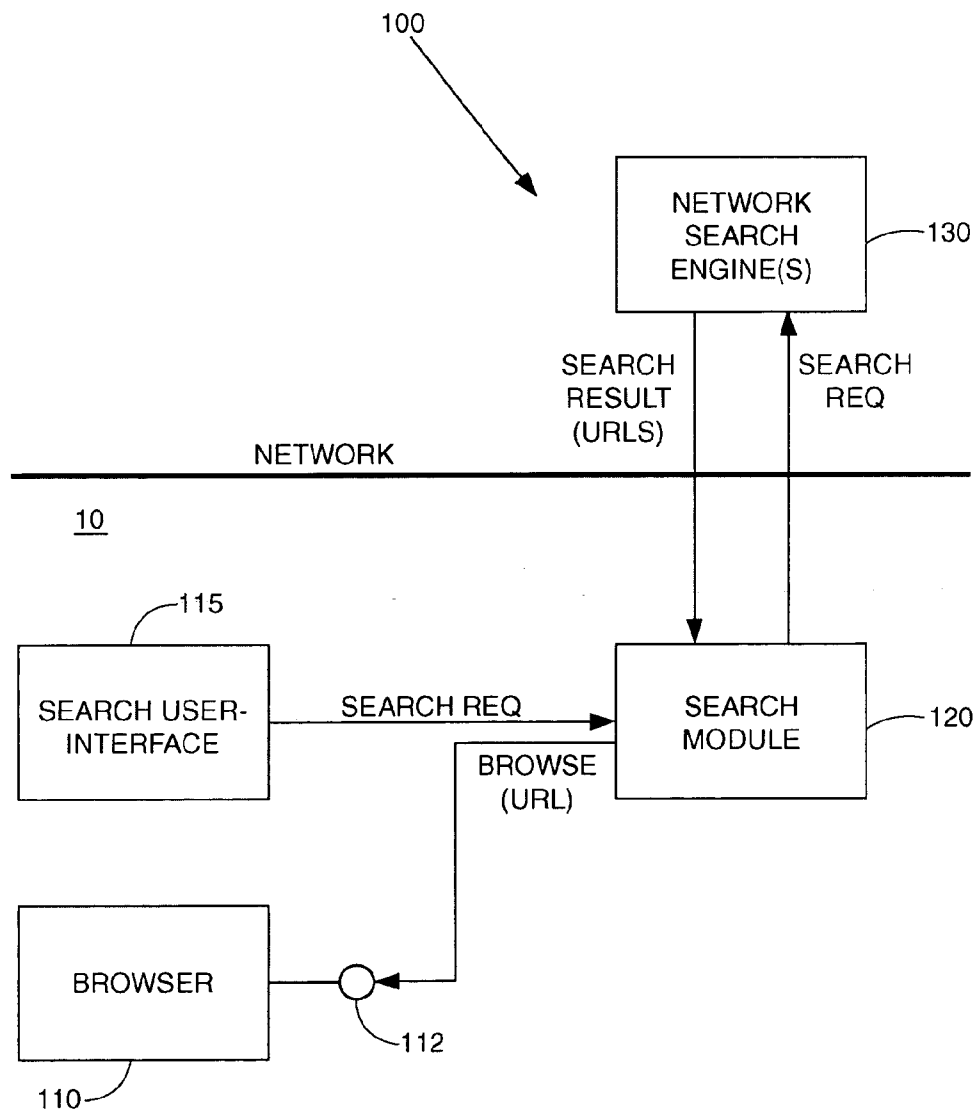
(73) Assignee: **Friskit, Inc.**, San Francisco, CA (US)

(57) **ABSTRACT**

(21) Appl. No.: **12/501,949**

An embodiment provides for conducting searches on a network. A search request is signaled over the network to a search engine. A search result is received that identifies a plurality of network addresses. Multiple pages are automatically rendered, each page being located by a corresponding network addresses in the search result.

(22) Filed: **Jul. 13, 2009**



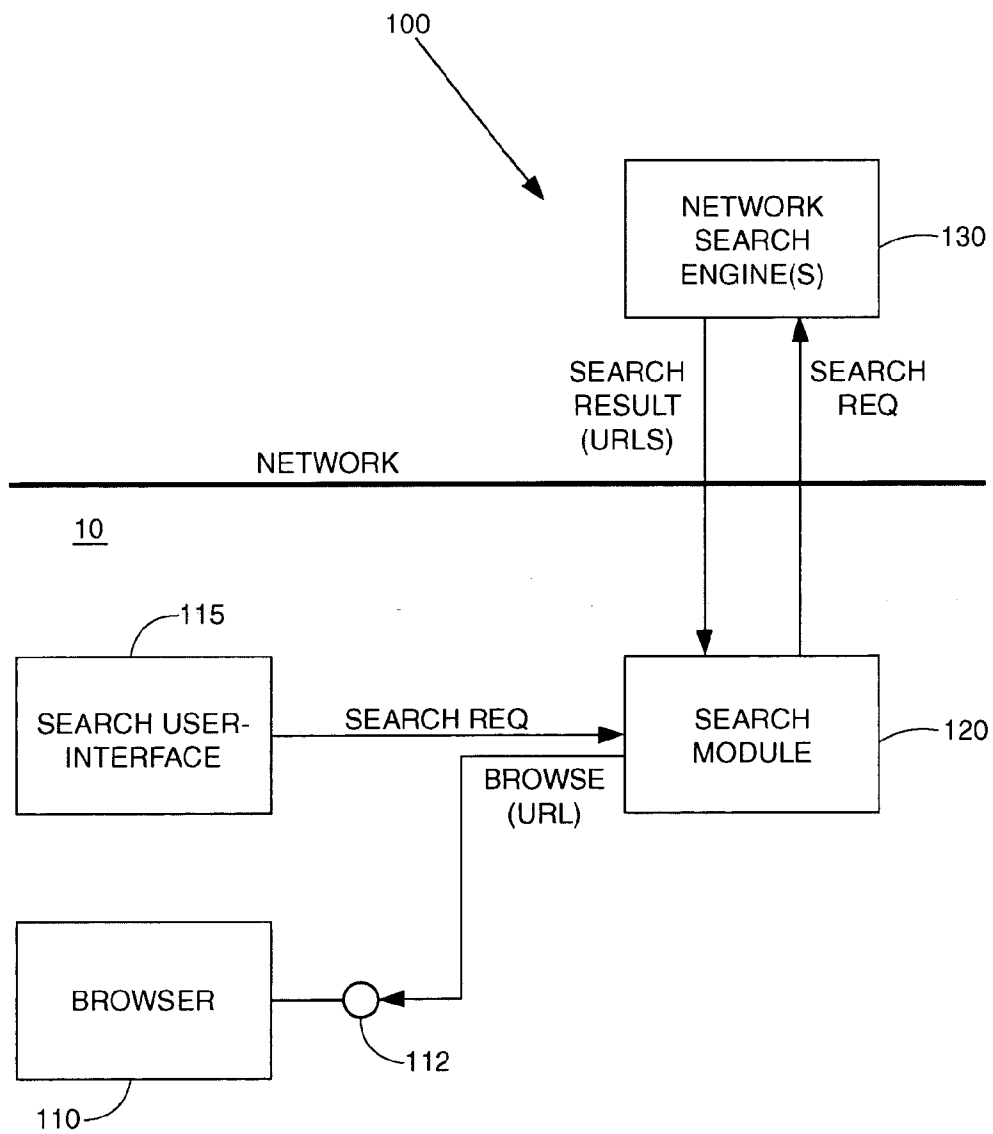


FIG. 1

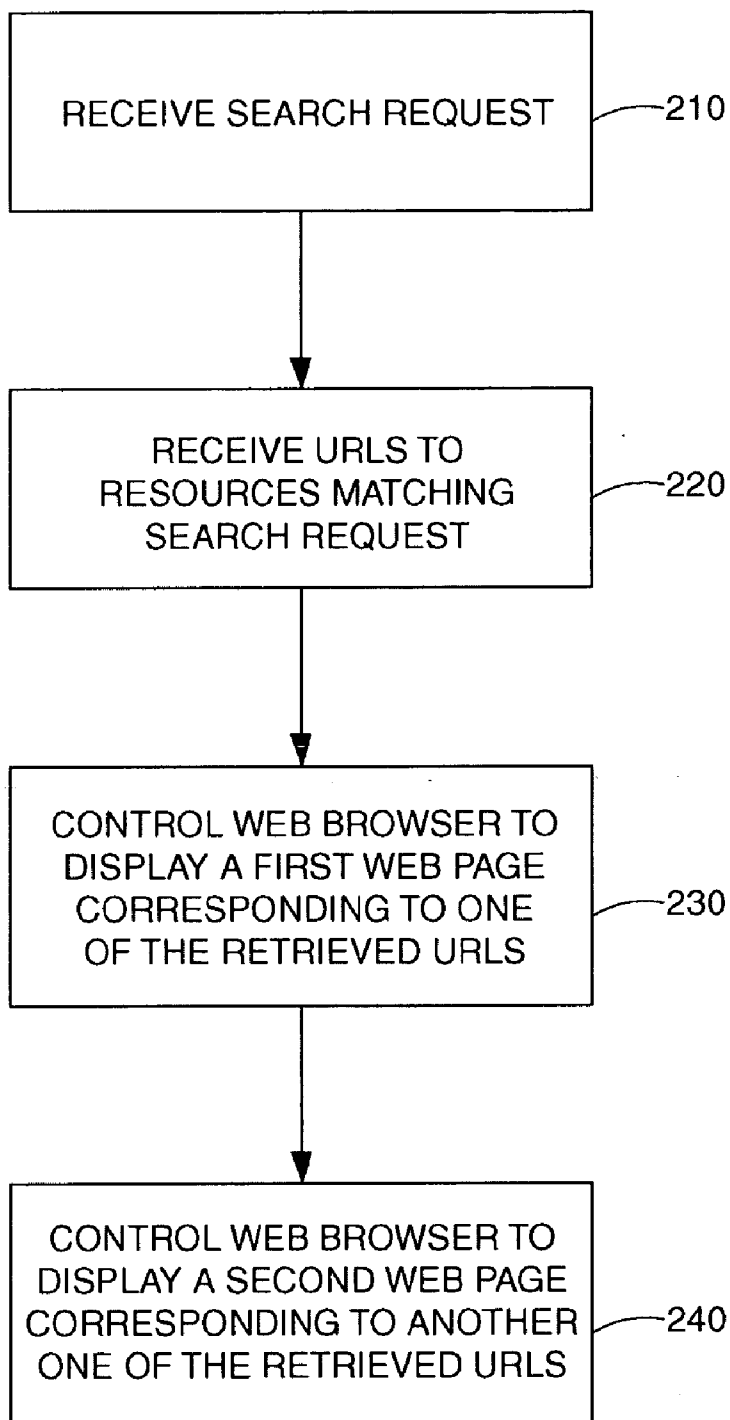


FIG. 2

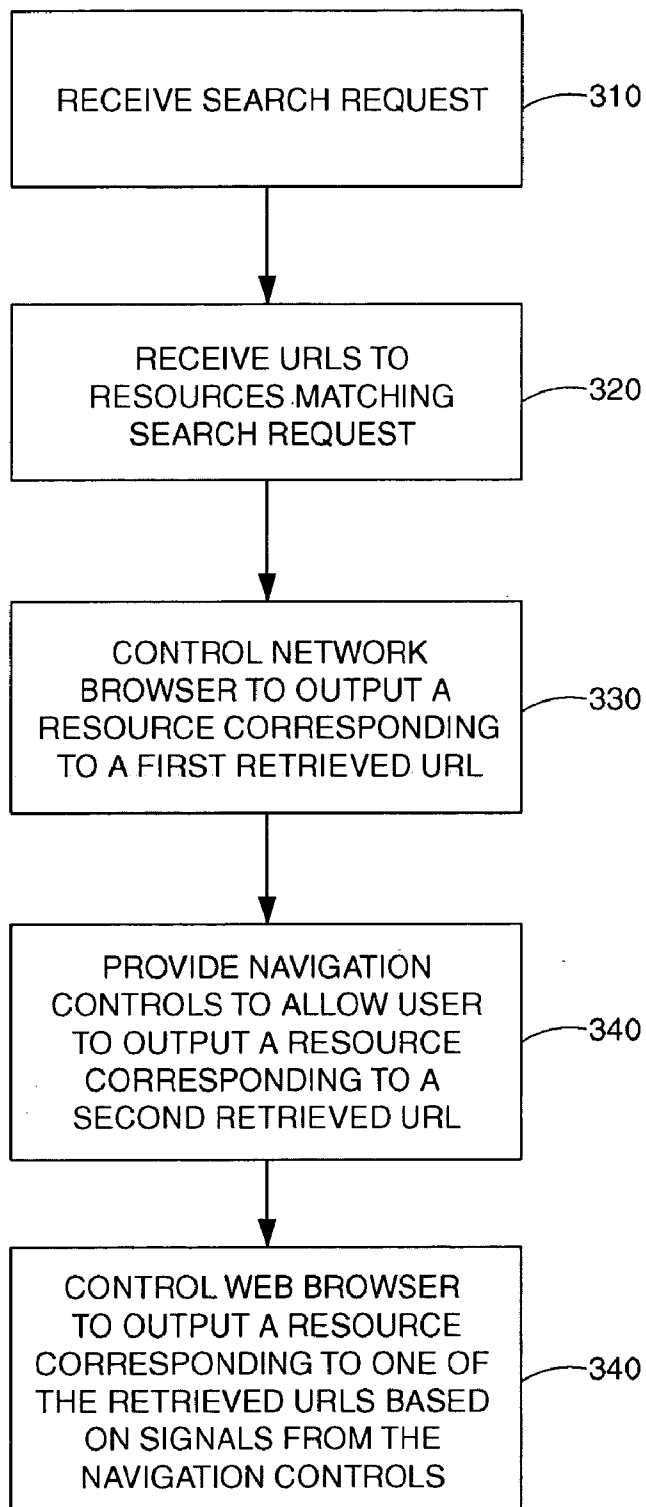


FIG. 3

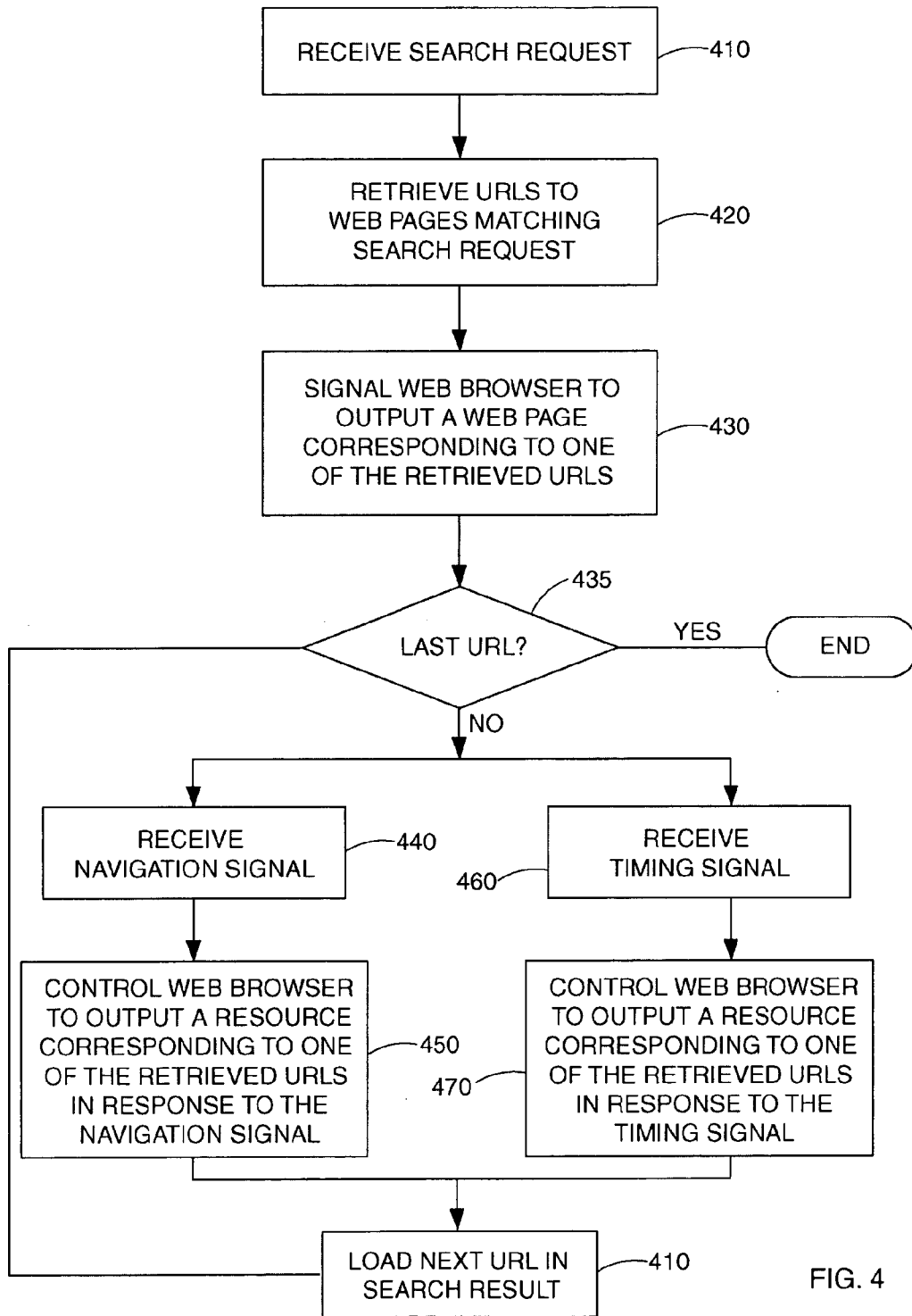


FIG. 4

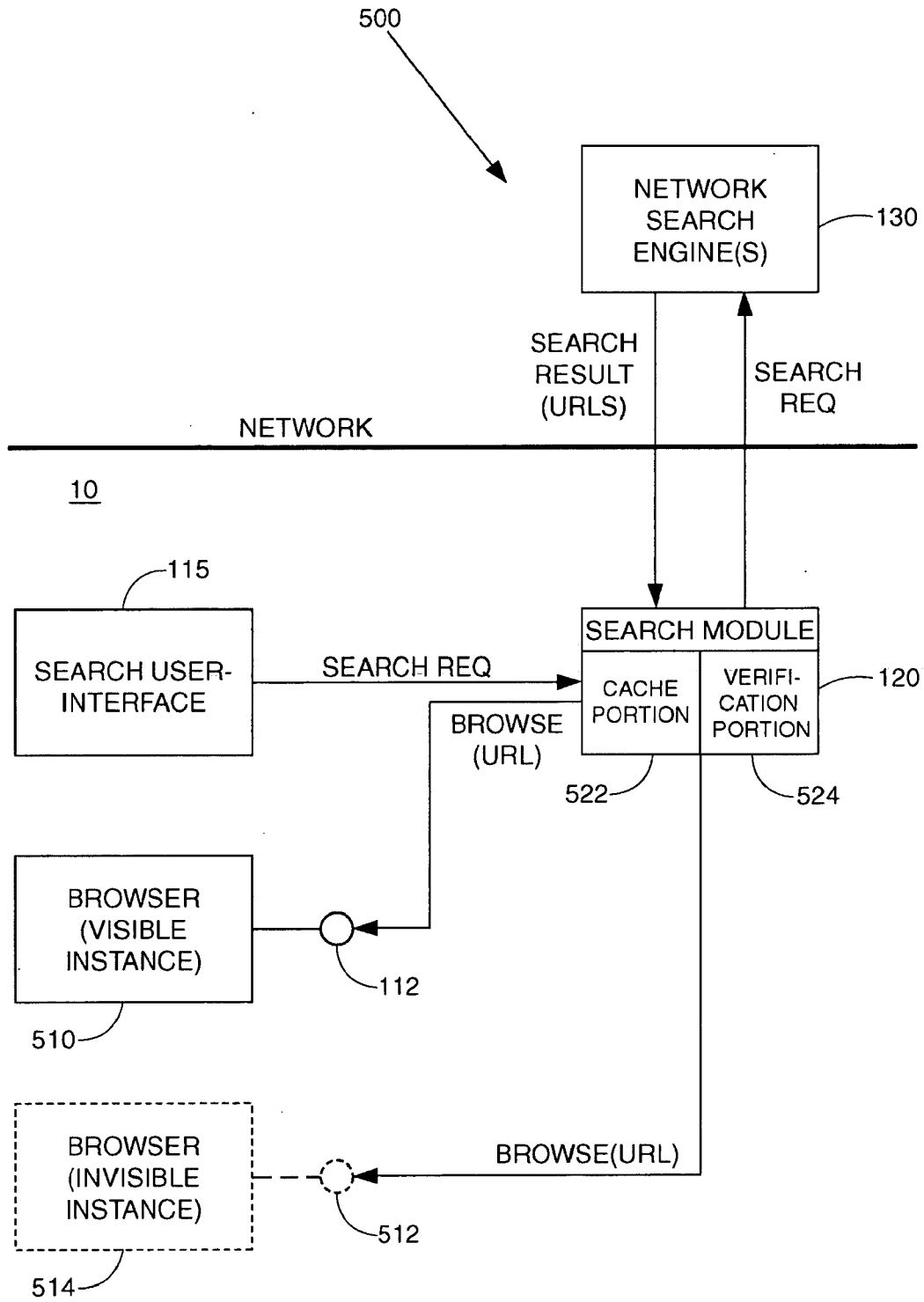


FIG. 5

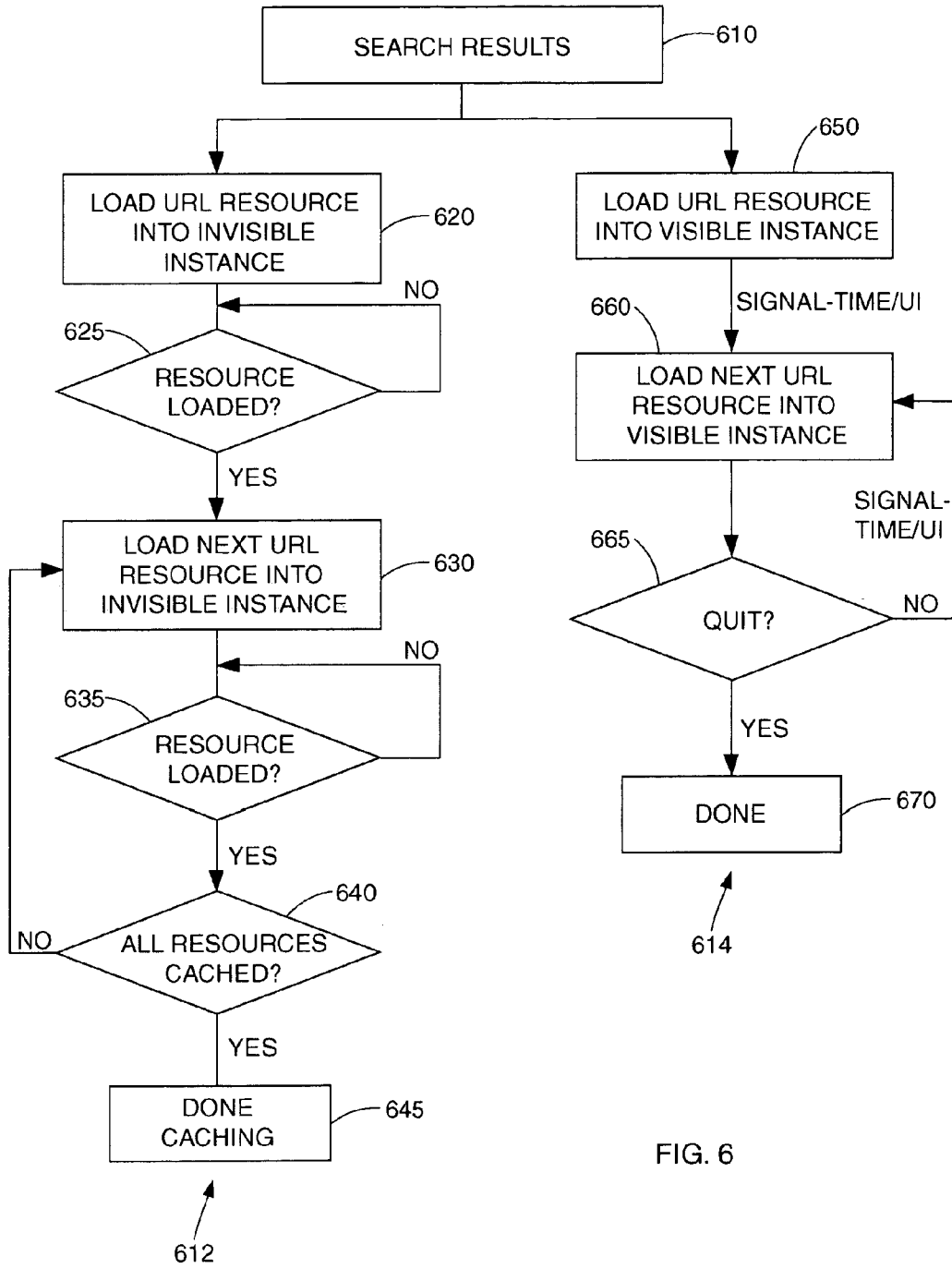


FIG. 6

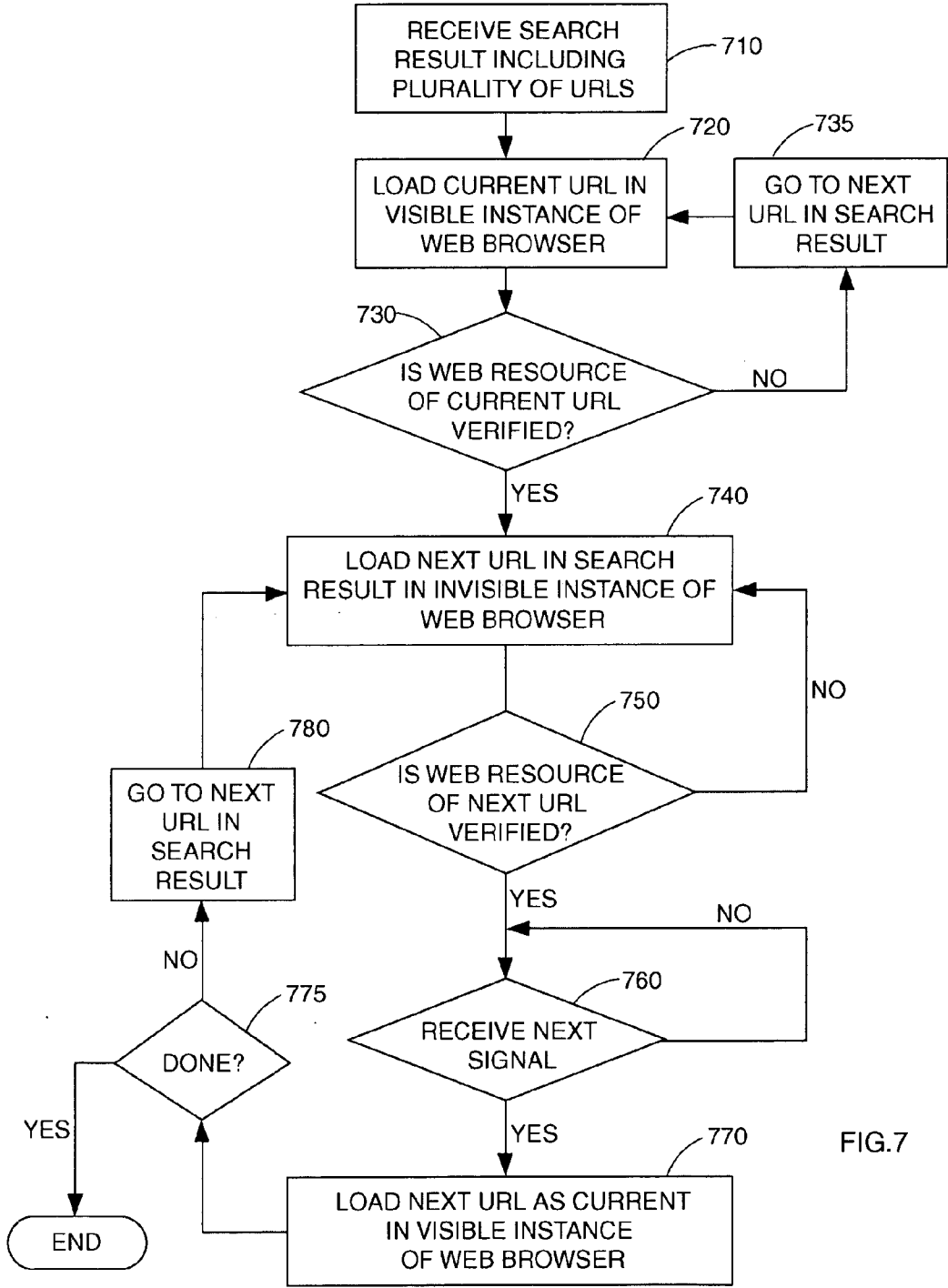


FIG. 7



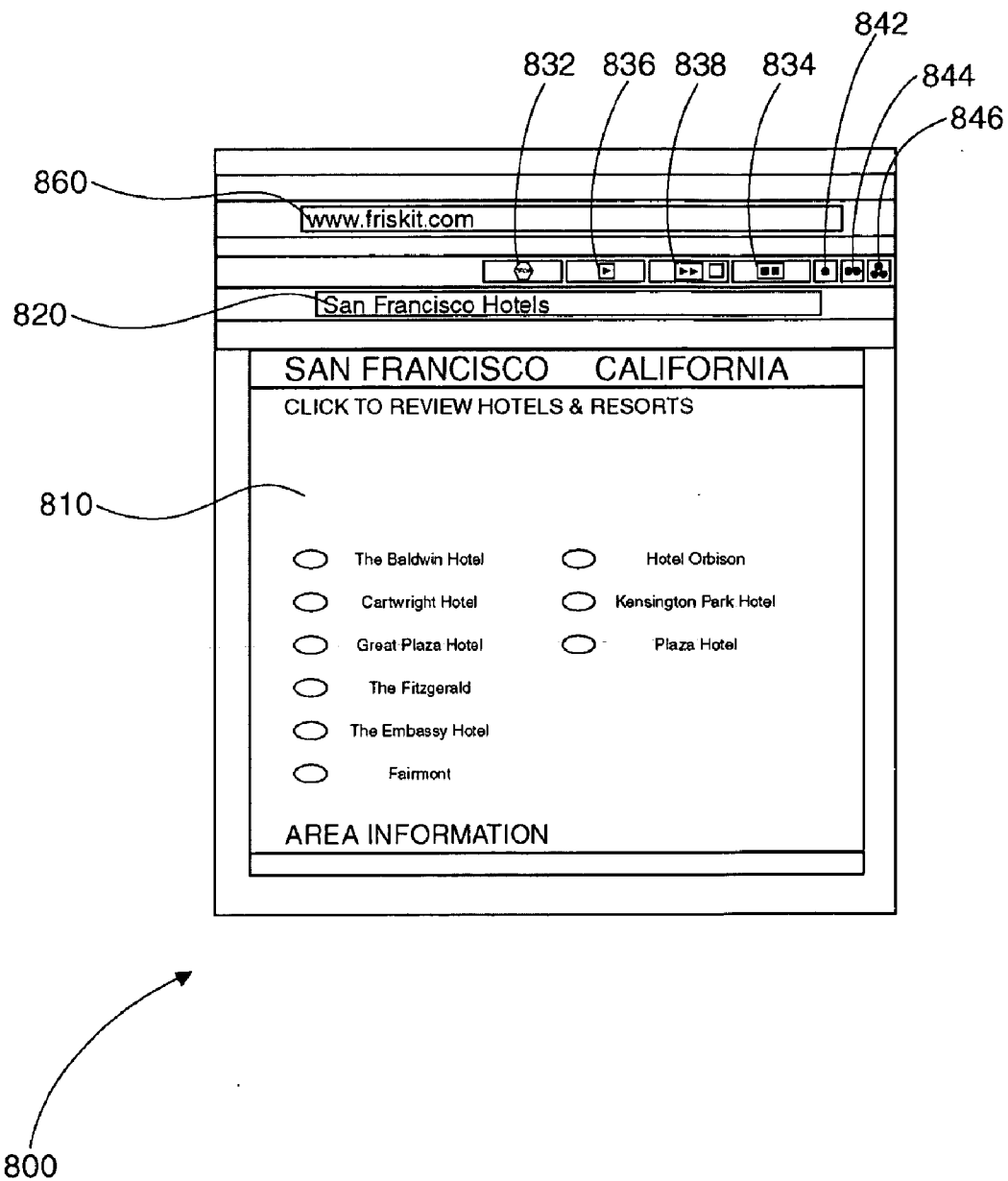


FIG. 8

**METHOD AND SYSTEM FOR VISUAL NETWORK SEARCHING**

**RELATED APPLICATIONS**

[0001] This application is a continuation of U.S. patent application Ser. No. 11/246,740, entitled "Method and System for Visual Network Searching," filed Oct. 7, 2005, which is a continuation of U.S. patent application Ser. No. 09/843,286, now issued U.S. Pat. No. 7,010,537, entitled "Method and System for Visual Network Searching," filed Apr. 25, 2001 and naming A. Eyal and S. Shor as inventors; which claims benefit of priority to U.S. Prov. Patent No. 60/200,716, entitled "Method and System for Visual Network Searching," filed Apr. 27, 2000 and naming A. Eyal and S. Shor as inventors; wherein all of the aforementioned priority applications are hereby incorporated by reference in their entirety and for all purposes.

**BACKGROUND OF THE INVENTION**

[0002] 1. Field of the Invention  
[0003] This invention relates to the field of searching for digital information on a network. In particular, the invention relates to network searching using visual feedback.  
[0004] 2. Background  
[0005] Search engines exist on the Internet to locate web sites that match a particular search criteria. Users on terminals may submit search terms and requests in order to receive results that are determined to match the search request. The results are usually provided to the user as a list of links. The user must select each link to view the corresponding page. When one page is opened, the user must select to return to the page having the search result to select another link. Thus, to view each link located by a search result, the user may have to make one or two selections, requiring multiple web pages to be loaded and/or displayed.

**SUMMARY OF THE INVENTION**

[0006] Embodiments of the invention allow users to search for web pages on the Internet, and to view search results in an animated fashion. In an embodiment, a search engine returns a response to a query of a user. Rather than provide the result as a compilation of selectable links, a page for a first link specified in the result is rendered for the user. A page for a next link may automatically be rendered after the page for the first link is displayed, creating a slide show effect. Alternatively, the user may be provided a user-interface for selecting to view a page for another link. The page for the other link is then displayed without requiring the viewer to actually select the link to that page.

**BRIEF DESCRIPTION OF THE FIGURES**

[0007] FIG. 1 illustrates a block diagram of a system, under an embodiment of the invention.  
[0008] FIG. 2 illustrates a method to display network sites in response to a search request, under an embodiment of the invention.  
[0009] FIG. 3 illustrates a method to display web pages in response to a search request, under an embodiment of the invention.  
[0010] FIG. 4 illustrates another method to display network sites in response to a search request, under an embodiment of the invention.

[0011] FIG. 5 is a system including verification and caching of URLs in a response to a search request, under an embodiment of the invention.  
[0012] FIG. 6 illustrates a method for a system such as shown in FIG. 5, where caching and displaying the search result are performed as independent processes, under an embodiment of the invention.  
[0013] FIG. 7 illustrates a method for a system such as shown with FIG. 5, under an embodiment of the invention.  
[0014] FIG. 8 illustrates a user-interface, under an embodiment of the invention.

**DETAILED DESCRIPTION**

**A. System Overview**

[0015] Embodiments of the invention display resources of information on network sites that match a search request. The response to the search request is made by way of displaying one or more network sites that match the search request. The network sites can be displayed in a sequential and animated manner, without requiring additional selection or interaction by the end user.  
[0016] Embodiments of the invention include several advantages over other known systems. In other systems, the search result includes links to network sites, listed in selectable form. The user is then required to select each link manually, and then recall the search results after viewing each link before selecting another link. In contrast, a user under an embodiment of the invention is able to view network sites in the search result, without having to select each link manually. Moreover, the user does not have to recall the search results to select each additional link in the search result, or to browse additional sites.  
[0017] In an Internet application, an embodiment of the invention displays one or more web pages in response to a search request. The feedback to the user may be in the form of a slide show, where web pages matching the search request are sequentially displayed to the user. The web pages may be displayed automatically. Alternatively, the user may be provided a navigation control on a user-interface to select web pages of other links in the search result.

**B. System Diagram**

[0018] FIG. 1 illustrates a block diagram of a system 100 for providing visual network content feedback in response to a search request, under an embodiment of the invention. The visual feedback may be in the form of displaying resources of the multiple network sites comprising the response. For example, the system 100 may automatically display web pages in a sequential manner to Internet sites that are determined to match a criteria set forth in the search request.  
[0019] In an embodiment, system 100 includes terminal and server side components that combine to provide the visual feedback. On the terminal, system 100 includes a network browser 110 and a search user-interface module 115. The network browser 110 includes an Application Programmable Interface (API) 112 that exposes its functionality. The search engine 120 communicates across a network with one or more network search engine(s) 130.  
[0020] In an embodiment, the network is the Internet. Other embodiments may be implemented on any network that carries digital information, such as local-area networks (LANs), Wide Area Networks (WAN), Extranets, Intranets, Internet, and wireless networks, or networks utilizing wireless trans-

missions. An example of a network for use with an embodiment of the invention includes a network operating under a transmission control protocol/Internet protocol (TCP/IP). Embodiments of the invention may also be employed on proprietary WANS, such as America Online™. Thus, discussion of embodiments employed on the Internet are exemplary, and equally applicable to other types of networks described above.

**[0021]** A user on user terminal **10** signals a search request using the search user-interface module **115**. The search request is received by search module **120**. The search request may be in the form of a text based entry. Alternatively, the search request may include a voice command. The user-interface **115** or search module **120** may interpret the voice command for the network search engine **130**.

**[0022]** The search module **120** signals the search request to a network search engine **130**. The search engine **130** may be configured for the network being used. In an embodiment, search engine **130** includes Internet web sites such as Yahoo®, Lycos®, and Infoseek®. The search request may be made to more than one network search engines. The system **100** is configurable to identify and retrieve only the best matching URLs in the results returned from each of the network search engines **130**. In an embodiment, the system **100** then uses only the best matching URLs to display to the user. In some embodiments, the network search engines include internal search engines located on specific web sites. For example, the network search **30** engine may be located on an e-commerce sites such as Amazon.com™ or Ebay™.

**[0023]** The network search engine **130** returns a result in response to the search request. The result is received by search module **120**. Unless no matches are identified, the result includes one or more URLs. Each URL in the result locates a network site having resources that match the search request, according to network search engine **130**. As an example, the network resources may correspond to text appearing on the network site, or identifiers used to identify the site with a search engine.

**[0024]** Once network search engine **130** responds, the search module **120** controls the network browser **110** to access and display the site corresponding to each URL in the search result. The search module **120** controls the network browser **110** through commands transmitted to the API **112**. In an embodiment, the search module **120** signals each URL with a command to network browser **110** so as to cause the network browser component to output a network resource of each URL in the search result. For example, the search module **120** signals the API **112** to <browse(URL)> for each URL in the search result.

**[0025]** The commands may be sent sequentially to for each URL in the search result. Thus, when the search result contains multiple URLs, the network resources are outputted sequentially by the browser **110**. For example, in Internet applications, a web page for each URL in the search result may be displayed in a sequential manner. The display of the web pages may be made to simulate a slide show.

**[0026]** Further, API **112** may be signaled to display web pages on the end terminal in a full-screen mode. Since the display of the search results is animated and requires no user input, the full-screen mode can be implemented without displaying control objects on the display.

### C. Flow Processes for Embodiments of the Invention

**[0027]** FIG. 2 illustrates a flow process for a system such as described with FIG. 1, operating on a network such as the

Internet. For discussion, network browser **110** is assumed to be a web browser, and components and features described with FIG. 1 are assumed to be adapted for the Internet.

**[0028]** In step **210**, search module **120** receives the search request from user terminal **10**. In response to receiving the search request, in step **220**, the search module **120** retrieves URLs to web resources that match the search request.

**[0029]** In step **220**, search module **120** may access a web search engine. The web search engine then identifies URLs to web pages that match the search criteria.

**[0030]** In step **230**, search module **120** controls the web browser **110** of user terminal **10** to display a web page of a first URL retrieved in step **220**. The first URL is signaled with control information to the API **112** of web browser **110**.

**[0031]** In step **240**, search module **120** controls the web browser **110** to display a web page of a second URL retrieved in step **120**. The web browser **110** may be controlled so as to display a web page of the second URL after the web page of the first URL is displayed.

**[0032]** In an embodiment, web pages of additional URLs contained in the search result may be displayed in a manner similar to web pages of the first and second retrieved URLs. Thus, the search result may be provided as multiple web pages that are displayed on, for example, a monitor of user terminal **210**. The web browser may be controlled by search module **120** to display the web pages in the search result in an animated manner.

**[0033]** Once all the web pages of URLs in the search result are displayed, the process in FIG. 2 may be repeated. Thus, the web pages may be redisplayed. This process may continue until the user signals otherwise. When the web pages are re-displayed, the web browser **110** has already cached the corresponding URLs. Thus, the display of the web pages may occur more quickly, or in a more continuous and animated fashion.

**[0034]** FIG. 3 is a flow process for another embodiment of the invention, employed with a network browser for networks including Internet, wide-area networks such as America Online™, and Intranets.

**[0035]** In step **310**, a search request is received by search module **120**. Then in step **320**, search module **120** retrieves URLs to network resources that match the search request. Each URL accesses a corresponding network resource. The network resources comprise data that forms an output on the user terminal **10**, once the network resources are accessed by the network browser **110**.

**[0036]** In step **330**, the search module controls the network browser **110** to output a resource corresponding to a first one of the retrieved URLs. The resource outputted may correspond to, for example, a web page.

**[0037]** In step **340**, a navigation control is provided to a user on the user terminal **10**. The user can select to output a resource corresponding to a second one of the retrieved URLs using the navigation control. An example of a navigation control is provided with FIG. 8. The navigation control may be provided by search user-interface module **115**. The navigation control may include a user-interactive feature that signals search module **120** to access a next URL from the retrieved URLs. The navigation control may also provide user-interactive features to replay a resource from one of the URLs, or to skip to a specific URL. The navigation control may also “pause” the output on user terminal **10** of a network resources corresponding to one of the retrieved URLs. Other navigation controls are described with FIG. 8.

[0038] In step 350, network browser 110 is controlled to output a resource corresponding to one of the retrieved URLs, based on a control signal received by the navigation controls. For example, a “next” signal may be provided through input to the navigation control, causing a network resource corresponding to a second retrieved URL to be outputted on the user terminal 10.

[0039] Additional network resources of other URLs contained in the search result may be provided by repeating steps 340 and 350. Once all URLs in the search response are outputted via network browser 110, the process may end. In an embodiment, the search module 120 may be configured to repeat signaling each URL in the search result to network browser 110 once all URLs in the search request are signaled. Thus, the process may be continuously repeated until the user signals to stop the output of the network resources.

[0040] In an embodiment, the order in which the network sites are displayed to the user are random. In another embodiment, the order in which the network sites are displayed to the user correspond to the order of matching each site is designated by the network search engine 130.

[0041] FIG. 4 illustrates a flow process under another embodiment of the invention. For illustration, the embodiment of FIG. 4 is assumed to operate with a terminal coupled to the Internet.

[0042] In this embodiment, the search request is received in step 410. In response, step 420 provides that a plurality of URLs are retrieved that have web pages determined to match the search criteria. In step 430, the web browser on user terminal 10 is signaled to output a web page corresponding to one of the URLs. The web page may be a site that contains one or more web resources matching the search criteria.

[0043] In step 435, a determination is made as to whether any other URLs exist among the retrieved URLs. If the determination is negative, the process is done.

[0044] In step 440, a navigation signal may be received through a navigation control interface. Then, in step 450, the web browser is controlled to output a web page corresponding to one of the retrieved URLs. This is done in response to the navigation signal being received in step 440.

[0045] In step 460, a timing signal may be actuated to retrieve a web page of a second URL from the retrieved URLs. For example, search module 120 may be programmed to provide the timing signal after one second. The timing signal may be configurable by the user on user terminal 10, using an interface such as described with FIG. 8. For example, the user may select the frequency at which each web page is displayed in response to a search request. In step 470, upon receiving the timing signal, the web browser on user terminal 10 is controlled to output a web page corresponding to another one of the retrieved URLs.

[0046] In step 480, a next URL among the retrieved URLs is loaded into web browser 110. The process returns to step 435.

[0047] In an embodiment, steps 460 and 470 are a default in case a navigation signal is not received. That is, unless the user on user terminal 10 instructs otherwise, the search module 120 will use a timing signal to determine when the next web page corresponding to a next URL in the retrieved URLs is displayed.

#### D. Embodiment Including Verification and Caching

[0048] FIG. 5 illustrates an embodiment in which a system 500 includes verification and caching functions. In this embodiment, web browser 110 includes a visible instance 510 and an invisible instance 514.

[0049] As with previous embodiments, a user enters a search request through search user-interface module 115. The search user-interface module 115 signals the search request to search module 120. The search module 120 forwards the search request to one or more search engines residing on the network. A search result comprising one or more URLs is provided from each of the network search engines used. The search module 120 then signals API 112 of visible instance 510 a command to browse each of the URLs in the search request. The search module signals the API 512 of the invisible instance 514 a command to browse the URLs in the search request as well. The relative sequence in how the visible instance 510 and the invisible instance 514 are signaled the URL are described in greater detail with FIG. 7.

[0050] The invisible instance 514 may be employed by search module 120 for purpose of verifying links and/or caching URLs returned in the search results. The search module 120 may communicate with the invisible instance 514 through a corresponding API 512.

[0051] A caching portion 522 of search module 120 caches the network resource of the URLs in the search request prior to the web browser displaying the network resources of the URLs. Under an embodiment of the invention, when the search results are returned by the network search engine 130, the visible instance 510 of web browser 110 loads a current URL from the search result. The visible instance 510 of web browser 110 then displays resources on the site of the URL using search user-interface module 115. Concurrently, the invisible instance 514 loads a next URL retrieved in the search request, so that the resource of the next URL, or next URLs, in the search result is cached before the network resource of the next URL is displayed.

[0052] A verification portion 524 of the search module 120 verifies that network resource identified by the next URL returned with the search result can be loaded to display or otherwise output network resources on the user terminal 110. When network resources of a current URL are loaded in the visible instance 510, the verification portion 524 loads the network resources of the next URL in the invisible instance 514. If network resources are returned for the next URL signaled to the invisible instance 514, the next URL is verified by the verification portion 524. In an embodiment, the next URL is verified if the network resource of the next URL is available and unbroken. Only the network resources of verified URLs are loaded by the visible instance 510.

[0053] The verification portion 524 and caching portion 522 occur independently of the network resources loaded in the visible instance 510. Thus, the caching portion 522 and verification portion 524 may cache and verify network resources of additional URLs while the visible instance 510 has loaded network resources of a previous URL.

[0054] In an embodiment such as shown by FIG. 4, the web browser 110 and search module 120 reside on the user terminal 10. However, in other embodiments, portions of the search module 120, such as the verification portion 524 reside on a server accessible to user terminal 10 through a network such as the Internet.

[0055] FIG. 6 illustrates another embodiment where caching and displaying URL resources identified by the search result occur concurrently and independently of each other. This process may be employed with a system such as described with FIG. 5. The system attempts to cache the web pages identified by the search request while web pages are individually being displayed on the end terminal. Thus, the

system may attempt to cache all the web pages located in the search while concurrently displaying web pages located by the search.

[0056] In step 610, the search result is returned from the network search engine 130. For purpose of description, the search result is assumed to contain a plurality of URLs. Once the search result is returned, the user terminal 10 concurrently performs a caching process 612 and a display process 614. For this embodiment, the caching and display process are performed independent of each other.

[0057] In the caching process 612, step 620 is performed to load a URL resource into an invisible instance 514 of the web browser. In step 625, a determination is made as to whether the URL resource was loaded. If the determination is negative, then step 620 is performed again. In step 630, the next URL resource in the search result is loaded into the invisible instance 514 of the web browser. In step 635, a determination is made as to whether the next URL resource was loaded. If the determination is negative, step 630 is repeated.

[0058] If the next URL resource was loaded, then in step 640 a determination is made as to whether all URL resources in the search result have been cached. If all resources have been cached, then the caching process is done in step 645. If additional caching is required, then step 630 is repeated.

[0059] In an alternative embodiment, caching may be terminated prior to all of the URL resources in the search result being displayed. This may occur if all of the search results are displayed, or if the display process 614 is stopped.

[0060] The display process 614 is performed independently of the caching process. In step 650, a URL resource from the search result is loaded into the visible instance 510 of the web browser. In step 660, the next URL resource identified by the search result is loaded into the visible instance 510. The next URL resource may be loaded upon the visible instance 510 being signaled to load the next URL resource. The signal to the visible instance 510 may be a timing signal. Alternatively, user input may cause the next URL resource to be loaded.

[0061] In step 665, a determination is made as to whether the display process is completed. If the determination is positive, then the display process 614 is complete in step 670. Otherwise, step 660 is repeated. The display process 614 may be complete if, for example, the display process 614 is timed out. Alternatively, the display process 614 may be complete if user input stops the display process 614. For example, the user may select an icon on the user-interface 800 (See FIG. 8) to stop an animated slide show displaying the URL resources identified by the search result.

[0062] In an embodiment, the caching process 612 is terminated once the display process 614 is complete. Also, in an embodiment, the display process 614 may continue after the caching process 612 is complete.

[0063] FIG. 7 illustrates a flow process for use with system 500, under an embodiment of the invention. The system 500 is assumed in this embodiment to operate on a network such as the Internet.

[0064] In step 710, the search result is returned from the network search engine 130. For purpose of description, the search result is assumed to contain a plurality of URLs. In step 720, a current URL is loaded into the visible instance 510 of web browser 110. In this step, the current URL corresponds to a first URL in the search request.

[0065] In step 730, a determination is made as to whether the current URL is verified. To be verified, a web resource has to be returned when the current URL is loaded into the web browser. For example, a web page of the URL has to be displayed on user terminal 10. The determination in step 730

may be negative if the URL is a broken link. The link may be broken if, for example, the web resource is no longer available. The URL may also be broken for other reasons, such as network congestion or failure by a hosting server of the URL. The determination in step 730 may also be negative if the web site of the URL is unavailable, such as in the case when there is traffic blocking access to the Internet. If the determination in step 730 is negative, a next URL in the search result is assumed to be the current URL in step 735. Then, step 720 is repeated.

[0066] If the determination in step 730 is positive, the current URL is loaded to be verified by the visible instance 510, and the web page (or web resource) of the URL is displayed to the user of user terminal 10. The search user-interface module 115 may be used to display the web page. Concurrently in step 740, the next URL in the search result is loaded into the invisible instance 514 of web browser 110.

[0067] In step 750, a determination is made as to whether the next URL is verified. The determination is made by the invisible instance 414 of web browser 110. If determination is negative, in step 740 is repeated for a next URL. If the determination in step 750 is positive, then the invisible instance 514 has received a web page (or web resource) from the next URL. Thus, the next URL is not a broken link, and has an available web page.

[0068] In step 760, a determination is made as to whether a signal has been received to load a next URL in the visible instance 510 and invisible instance 514 of web browser 110. In an embodiment, the signal is a timing signal from the search module 120. For example, after a predetermined number of seconds, the timing signal may signal to load the next URL after a few seconds. In other embodiments, the signal is provided by a user of user terminal 10. The user signal may be provided through a navigation control 730, such as described with FIG. 8.

[0069] If the determination in step 760 is positive, in step 770 the next URL previously loaded in the invisible instance 514 is loaded in the visible instance 510. In step 775, a determination is made as to whether the URL loaded in the visible instance 510 is the last URL in the search result. If so, then the process is done.

[0070] If there is another URL in the search result, the system 500 goes to the next URL in step 780. The flow process is repeated for the next URL in step 740.

#### E. User-Interface

[0071] FIG. 8 illustrates a user-interface 800 for user terminal 10, under an embodiment of the invention. For purpose of description, user-interface 800 is described with respect to Internet applications. The user-interface 800 may occupy a portion of a screen or monitor of user terminal 10. A portion of the screen or monitor may be dedicated to displaying network resources of retrieved URLs.

[0072] In an embodiment, the user-interface 800 cooperates with web browser 110 on user terminal 10. The user-interface 800 includes a display portion 810 that displays web pages of retrieved URLs. A search field 820 receives a search request as text entry. For example, key words such as "San Francisco" may be entered into the search field 820. Once the search request is entered, the system 100 returns URLs that match the search request to the web browser of user terminal 10. Then the web browser sequentially accesses the web sites corresponding to the URLs. The web browser displays a web page for each site before accessing the next site. The web page for each site is displayed in display portion 810 of user-interface 800.

[0073] The user-interface 800 includes a navigation control 830 to allow a user to navigate the display of web pages. In an embodiment, the web browser 110 is programmatically controlled to display web pages sequentially, in an animated fashion. For example, web pages to all of the web sites returned in the search may be displayed in the manner of a slide show, where one web page is displayed, then a next, until of the web pages in the search are shown. In an embodiment, a next web page may be displayed a predetermined time duration after a previous web page is displayed on the display portion 810.

[0074] In an embodiment, the navigation control 830 is configured to provide controls for a slide show. The navigation control may include a stop icon 832 that stops the that stops the slide show at a selected web page. The navigation control 830 may include a pause icon 834 that can be actuated to pause the slide show at the selected web page. Re-actuation of the pause icon 834 causes the slide show to continue from the selected web page. The navigation control 830 may also include a play icon 836 to cause the web browser to start displaying web pages in the manner of a slide show. A skip icon 838 causes the web browser to load a next URL retrieved in the search, and to display the web page for the next side. The skip icon 838 may be used to create the effect that a next slide was selected in the slide show. As mentioned in the embodiment of FIG. 4, a timing signal may serve as the default in causing the next web page to appear, unless the skip icon 838 is selected by the user. Alternative embodiments may use and arrange icons as manual controls, such as found on remote controls or VCRs.

[0075] The user-interface 800 may also include a web browser portion 860. The web browser portion 860 may be used to display the URLs of the web pages being displayed or loaded by the web browser component. Various web browser features may be provided with the web browser portion.

[0076] The user-interface 800 may also include a plurality of configuration fields. The configuration fields may be in the form of pull-down menus. A search selection menu 842 enables the user to select the search engine that matches the user's preference. For example, the user may configure the user-interface 800 to display specific search engines according to the user's preferences. A search type menu 844 enables the user to select the data type of the web resources for the search. For example, the user may select audio data type for music, or MPEG data types for specific type of music. A configuration menu 846 enables a user to configure the manner in which the web pages for the retrieved URLs is displayed. For example, the configuration menu may provide for a selectable item that configures the timing signal, so that the web pages are displayed faster or slow to the user.

F. Applications for Embodiments of the Invention

[0077] Embodiments of the invention may be implemented in applications such as described below.

[0078] In one application, an e-commerce site containing an internal search engine is contacted by user terminal 10. The search module 120 receives the search request specific for the e-commerce site, and then signals the search request to the internal search engine of the e-commerce site. For example, the search request may be for a specific product or service, such as books, music, and travel packages. The search result is then signaled to search module 120, which outputs web pages for each search result. Thus, a user may visually see prices and purchasing information for a product or service in response to a search request.

[0079] As an example, a user may enter an author name. The search module 120 signals the internal search engine of

the e-commerce site. The search result contains URLs to books by that author. The URLs are then sequentially and automatically displayed to the user. Thus, the user can see the various books, along with purchasing information such as pricing and availability, in rapid succession, without having to select and reselect links to navigate between different products.

[0080] As another example, the e-commerce site may be an auction site. The internal search engine of the auction site may return various auctions that match the search criteria. The user is able to see each auction sequentially, without having to select and reselect links to each auction individually.

[0081] Travel packages may be displayed to the user in a similar manner. Thus, users may submit a search request to rapidly see pictures of resorts, along with pricing information and availability.

[0082] As another application, a message board may include an internal search engine. The search request may specify content of messages. Each message matching the search request may be automatically and sequentially displayed to the user. Further, the user may configure the system to display messages for a certain duration, affording the user to read or skim each message. The navigation controls of search user-interface may be used to add further controls to the user. Another application may display message board threads to the user sequentially and in rapid succession.

G. Other Features and Alternative Embodiments

[0083] In other embodiments, the network resources corresponding to the first one and/or the second one of the retrieved URLs may be an audio output, or a combination of an audio output and a video output. The audio or video output may be in addition or combination with other embodiments described above in which the resources are displayed items on the network.

[0084] In another embodiment, the user-interface 700 includes a record user-interactive feature. The record feature may be selected to create a list of network sites that are of particular interest to the user. For example, a record icon may be selected once a web page is displayed, causing the URL web page to be stored in a favorite list. The favorite list may subsequently be selected to display all or some of the web pages therein. In an embodiment, the recorded list may be selected so that the network resources of the recorded URLs are replayed as a slide-show, or otherwise in an animated fashion.

[0085] In a variation, the user may select to add a URL of a web page being displayed to a bookmark of the web browser.

[0086] In another embodiment, the system 100 provides an audible sound for each network site displayed.

H. Conclusion

[0087] The foregoing description of various embodiments of the invention has been presented for purposes of illustration and description. It is not intended to limit the invention to the precise forms disclosed. Many modifications and equivalent arrangements will be apparent.

What is claimed is:

- 1. A method according to embodiments described herein.
2. A system according to embodiments described herein.

\* \* \* \* \*