(54) Title: METHOD AND APPARATUS FOR COMBINING VIDEO DATA WITH STATIC WEB PAGE DATA

(57) Abstract

What is disclosed is a method for presenting a video image (506) and compositing the video image (506) with a static web page in a browser application (50) including receiving a request from a user to initiate the playback of the video image (506) stored on a pre-distributed medium. Further, the disclosed method also includes transferring, from a network, a control file containing location information for the video image (506) and a vector-image-mapping (504) file associated with the video image (506). Processing the vector-image-mapping (504) file to retrieve a set of active areas (508) coordinates. Then associating the set of active areas coordinates with a set of areas in the video image (506); and, presenting the video image (506) while determining user interactivity within the set of areas in the video image. An apparatus for performing the method is disclosed.
FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<table>
<thead>
<tr>
<th>Code</th>
<th>State or Country</th>
<th>Code</th>
<th>State or Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>Albania</td>
<td>ES</td>
<td>Spain</td>
</tr>
<tr>
<td>AM</td>
<td>Armenia</td>
<td>FI</td>
<td>Finland</td>
</tr>
<tr>
<td>AT</td>
<td>Austria</td>
<td>FR</td>
<td>France</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
<td>GA</td>
<td>Gabon</td>
</tr>
<tr>
<td>AZ</td>
<td>Azerbaijan</td>
<td>GB</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>BA</td>
<td>Bosnia and Herzegovina</td>
<td>GE</td>
<td>Georgia</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
<td>GH</td>
<td>Ghana</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
<td>GN</td>
<td>Guinea</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
<td>GR</td>
<td>Greece</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
<td>HU</td>
<td>Hungary</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
<td>IE</td>
<td>Ireland</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
<td>IL</td>
<td>Israel</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
<td>IS</td>
<td>Iceland</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
<td>IT</td>
<td>Italy</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
<td>JP</td>
<td>Japan</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
<td>KE</td>
<td>Kenya</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
<td>KG</td>
<td>Kyrgyzstan</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d'Ivoire</td>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
<td>KR</td>
<td>Republic of Korea</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
<td>KZ</td>
<td>Kazakhstan</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
<td>LC</td>
<td>Saint Lucia</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
<td>LI</td>
<td>Liechtenstein</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
<td>LK</td>
<td>Sri Lanka</td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
<td>LR</td>
<td>Liberia</td>
</tr>
<tr>
<td>LS</td>
<td>Lesotho</td>
<td>LT</td>
<td>Lithuania</td>
</tr>
<tr>
<td>LU</td>
<td>Luxembourg</td>
<td>LY</td>
<td>Latvia</td>
</tr>
<tr>
<td>MC</td>
<td>Monaco</td>
<td>MD</td>
<td>Republic of Moldova</td>
</tr>
<tr>
<td>MG</td>
<td>Madagascar</td>
<td>MK</td>
<td>The former Yugoslav Republic of Macedonia</td>
</tr>
<tr>
<td>ML</td>
<td>Mali</td>
<td>MN</td>
<td>Mongolia</td>
</tr>
<tr>
<td>MR</td>
<td>Mauritania</td>
<td>MW</td>
<td>Malawi</td>
</tr>
<tr>
<td>MX</td>
<td>Mexico</td>
<td>NE</td>
<td>Niger</td>
</tr>
<tr>
<td>NL</td>
<td>Netherlands</td>
<td>NO</td>
<td>Norway</td>
</tr>
<tr>
<td>NZ</td>
<td>New Zealand</td>
<td>PL</td>
<td>Poland</td>
</tr>
<tr>
<td>PT</td>
<td>Portugal</td>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td>RU</td>
<td>Russian Federation</td>
<td>SD</td>
<td>Sudan</td>
</tr>
<tr>
<td>SE</td>
<td>Sweden</td>
<td>SG</td>
<td>Singapore</td>
</tr>
<tr>
<td>SI</td>
<td>Slovenia</td>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>SN</td>
<td>Senegal</td>
<td>SZ</td>
<td>Swaziland</td>
</tr>
<tr>
<td>TD</td>
<td>Chad</td>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>TJ</td>
<td>Tajikistan</td>
<td>TM</td>
<td>Turkmenistan</td>
</tr>
<tr>
<td>TR</td>
<td>Turkey</td>
<td>TT</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td>UA</td>
<td>Ukraine</td>
<td>UG</td>
<td>Uganda</td>
</tr>
<tr>
<td>US</td>
<td>United States of America</td>
<td>UA</td>
<td>United States of America</td>
</tr>
<tr>
<td>UZ</td>
<td>Uzbekistan</td>
<td>VN</td>
<td>Viet Nam</td>
</tr>
<tr>
<td>YU</td>
<td>Yugoslavia</td>
<td>ZW</td>
<td>Zimbabwe</td>
</tr>
</tbody>
</table>
METHOD AND APPARATUS FOR COMBINING VIDEO DATA WITH STATIC WEB PAGE DATA

Field of the Invention

The present invention relates to the field of use of presenting multimedia information using a computer network. More particularly, the present invention relates to presenting distributed multimedia information.

Description of Related Art

Currently, content providers who wish to provide multimedia data such as video and audio data to users over a network such as the Internet must be cognizant of the size of the multimedia data to be transferred as most users will not be willing to endure the length of time required to transfer large files. In addition, transferring large files poses other problems in cases where users do not have a stable network connection. This leads to interrupted information transfer, which in turn results in wasted bandwidth and time to perform re-transfers.

For example, assuming no control information overhead and perfect line conditions, a file that is one megabyte (MB) in size will take approximately 5 minutes to transfer using a modem having a transfer speed of 28,800 bits of second. However, under normal conditions, accounting for overhead data and imperfect line conditions, the transfer often takes much longer. With multimedia data files becoming larger in order to contain the length of materials desired, including book audio and video data, the time required for transferring these files are also becoming larger.

Although advances in communications networks have been made to increase the transfer speed of these files, the rate at which files are growing is still outpacing the rate at which transfer speeds are increasing. Consequently, content providers must design “web-sites,” which are comprised of files containing hyper-text markup language (HTML), digital audio and video information, so as not require a significant amount of time or network resources to load. Unfortunately, this means that content providers have to use lower quality versions of multimedia files, which has been subjected to some data compression schemes, to present their messages.

The network constraints described above are not acceptable for applications where large multimedia files must be transferred. For example, in a web-site
containing various advertisements for cruise ship utilizing audio/video information in addition to text, a content provider might desire to play a sample video for each cruise package in which a user expresses an interest. However, with the current level of network technology that is available to a majority of users, this desire is not achievable.

A proposed solution is to pre-distribute the multimedia data using removable media to an interested user which will allow the user to have any necessary information locally and thus by-pass a download of large multimedia data files. The user can then view video or listen to audio in conjunction with using a browser application to access a content provider's web-site without having to wait for the multimedia to be transferred over the network. However, a solution is necessary to provide integration of the display of the local multimedia information with the information that is displayed in the browser application.

**SUMMARY OF THE INVENTION**

The present invention utilizes custom software in a computer system having a processor, memory, a removable media data storage device, a graphics card, a display, and a network interface to allow the presenting of multimedia information which is contained on removable media and is triggered based upon the actions of the user using a browser application in viewing a content provider's website.

The custom software is installed into the computer system from either the network (e.g., from the content provider's server) or the removable media storage device. The custom software allows the user to access the databases and services of a content provider, and to trigger the presentation of multimedia information (such as audio or video data) based on the user's actions of "browsing" the content provider's website. In addition, the user is allowed to select regions in a presented video to trigger additional actions, such as presentation of other multimedia information or further browsing/accessing of a content provider's website. The multimedia information resides on removable media which is distributed to the user and placed by the user into the removable media storage device.

By distributing the removable media containing the multimedia information to the user before the user accesses the content provider’s website, the user would not have to transfer the multimedia information from the content provider’s website. Instead, the multimedia information will be accessible locally.
In one embodiment, the content provider is a travel agent having a server on the Internet providing users with hyper text mark-up language (HTML) documents. A user would be able to select links in a world-wide-web (WWW) browser application to receive tour information from the server and to trigger the presentation of multimedia data contained on the removable media. The removable media could be media such as compact disk read-only memory (CD-ROM) or digital video disc (DVD) media.

The multimedia information on the media allows a user to preview particular tour packages of interest. The user may also access one or more corresponding databases for volatile information such as tour prices, and availability. When the user has decided which tour they intend to schedule, the user may finalize the transaction by actually booking the tour using the travel agent’s scheduling system which will make the appropriate scheduling arrangements with a tour provider.

In another embodiment, the multimedia information can also allow a user to preview movies or other entertainment information. The user may then select links or hot spots on the presented multimedia information. For example, a user may select on an actor displayed in the present multimedia to find out more information about that actor in a new browser window.

**BRIEF DESCRIPTION OF THE DRAWINGS**

**Figure 1** is a block diagram of a network configured in accordance with the present invention;

**Figure 2** is a block diagram of a computer system configured in accordance with the present invention;

**Figure 3a** illustrates a memory of a computer system configured in accordance with the present invention to contain a player application.

**Figure 3b** illustrates the memory of **Figure 3a** configured in accordance with the present invention to include a plug-in for a browser application.

**Figure 4** is a flow diagram of one mode of operation of the present invention, including a multimedia file player.

**Figure 5** is a flow diagram of one mode of operation of the present invention, including the plug-in and the browser.
DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a method and apparatus for presenting distributed multimedia information. For purposes of explanation, specific embodiments are set forth to provide a thorough understanding of the present invention. However, it will be understood by one skilled in the art, from reading this disclosure, that the invention may be practiced without these details. Further, although the present invention is described through the use of the internet and the world wide web, most, if not all, aspects of the invention apply to networks and client/server applications in general. Moreover, well-known elements, devices, process steps and the like are not set forth in detail in order to avoid obscuring the present invention.

Figure 1 illustrates a network configured in accordance with the present invention, including a computer system 12, a network 14, and a server 16. Aside from the inventive portions of computer system 12, as discussed below, computer system 12 represents a general purpose computer which is configured to communicate with other computers such as server 16 over network 14. Similarly, server 16 represents a general purpose computer configured to communicate with other computers over network 14 to provide text and other data to client computers such as computer system 12.

In a preferred embodiment, network 14 is a wide-area network such as the Internet. However, network 14 can be of any type of network, or even be composed of several heterogeneous networks. Therefore, in other embodiments, network 14 can be a local area network, or a combination of local and wide area networks.

In the preferred embodiment, server 16 is a hyper-text transport protocol (HTTP) server that can provide text and other data to the various computer systems connected to network 14. Server 16 can also be configured to perform as a file transfer protocol (FTP) server which allows clients to access files located on the server.

Figure 2 illustrates a preferred embodiment of computer system 12, including a processor 22 coupled to a memory 24 over a memory bus 26 and a display device 28 coupled to processor 22 over a system bus 30. System bus 30 is also coupled to a cursor control device 32, a keyboard 34, a data storage device 36, and a network interface 38.

Computer system 12 includes system bus 30 for communicating information, and processor 22 is coupled to system bus 30 for processing information. Memory 24 can be random access memory (RAM) or another dynamic storage device
and is coupled to system bus 30 for storing information and instructions to be executed by processor 22. Memory 24 may also be used for storing temporary variables or other intermediate information during execution of instructions by processor 22.

Data storage device 36 is coupled to system bus 30 for storing information and instructions and can be a device such as a magnetic disk or optical disk data storage device, and is coupled to the other components of computer system 12 through the use of system bus 30. In the preferred embodiment, data storage device 36 is a removable media device such as a compact disk-read only memory (CD-ROM) drive or a digital versatile disc (DVD)/DVD read only memory (DVD-ROM) drive. The removable media could be distributed to users by mail or other means. Therefore, a user would have the removable media available to be placed into the removable media device at the appropriate time, as explained below. It is to be noted that data storage device 36 can be composed of several data storage devices instead of only the removable media device. Thus, in alternative embodiments data storage device 36 can represent a multiple disk array of “hard disks” (i.e., magnetic disk media), removable media devices, and other data storage devices.

Display device 28, which is a device such as a cathode ray tube (CRT), is also coupled to the other components in computer system 12 via system bus 30, for displaying information to a computer user. Keyboard 34, which is an alphanumeric input device including alphanumeric and other keys, is typically coupled to system bus 30 for communicating information and command selections to processor 22. Another type of user input device is cursor control device 32, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 22 and for controlling cursor movement on display device 28. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), which allows the device to specify positions in a plane.

Operating system 56 is a general purpose operating system which works with computer system 12 to provide the necessary system for the other software components discussed herein. Other than what is necessary to provide the functionality needed in the present invention, operating system 56 is similar to other operating systems of the prior art.

Figure 3a is a block diagram of one embodiment of memory 24 of computer system 12 containing a ULI player 52, a media command interrupter
(MCI)/video control library system layer 54, and an operating system 56. The contents of memory 24 can be loaded from data storage device 36 or retrieved from a server over network 14. **Figure 3b** is a block diagram of another embodiment of memory 24 of computer system 12 additionally containing a browser application 50 and a ULI plug-in 58.

In the preferred embodiment of the present invention, browser application 50 interprets and displays files coded in hyper-text mark-up language (HTML). Browser application 50 also initiates external procedure calls to operating system 56 and other applications such as ULI player 52. The functionality of browser application 50 can be extended by using a plug-in such as ULI plug-in 58, which is loaded at execution time.

**ULI player**

ULI player 52 is used to control the retrieval and presentation of multimedia information contained in data storage device 36 or other data sources. In addition to further functions as described below, ULI player 52 contains the following functions: VOLUME, VERSION, DISK, OPEN, EXECUTE, PUT_WINDOW_AT, and PLAY.

The VOLUME function allows ULI player 52 to search data storage device 36 for a removable media with a particular label, the label being a series of alphanumeric characters, a series of bits. Thus, for example, ULI player 52 would be able to look for the correct volume identification in a system with multiple volume.

The VERSION function allows ULI player 52 to check to see if the removable media is the correct version. This is to ensure that ULI player 52 would only play the most recent version of the multimedia files.

ULI player 52 provides the DISK function to check and see if the removable media currently being accessed has the correct identification number.

To play a multimedia file, ULI player 52 provides an OPEN function to open a multimedia file to send to MCI system layer 54.

ULI player 52 also uses the PUT_WINDOW_AT function to display a window in which a multimedia file will be presented, and passes the window frame information to MCI system layer 54 so that MCI system layer 54 will know where to display the video content of the multimedia file.

To actually initiate the playing of a multimedia file, ULI player 52 has a PLAY function, which instructs MCI system layer 54 to begin playing the multimedia file.
by displaying the video portion in a window that has been displayed by the
PUT_WINDOW_AT function. ULI player 52 is able to search data storage device 36
and any other data storage device which is coupled to computer system 12 to find the
necessary multimedia files to present. ULI player 52 is also able to verify that the file to
be presented is the appropriate version.

Other functions supported by ULI player 52 includes the following. A CAPTION
function allows the ULI player to change the caption bar of the web browser player
window to any given text. A REDIRECT function redirects the web browser to a URL
when the player is done playing. A BASE_URL function forces the web browser to be
on a specific web page before the ULI player starts playback of a multimedia file. A
START_OFFSET function starts the media at a given frame number and allows the
media. A WINDOW_SIZE function forces the player window to be a given size and
position while FULL_SCREEN function forces the player into full screen mode. An
AUTO_START function automatically starts playing the media. A LOOP function
restarts the media after it is done playing. A CONTROLS_ON function turns on the
player controls while a CONTROLS_ALLOWED function disables or enables the ability
to turn controls on or off. A VERSION function requires a specific version of the CD
inserted before the media can be played. An ID function requires a specific ID for the
inserted CD to be found before the media can be played.

ULI plug-in

ULI plug-in 58, as described above is a plug-in that works with browser
50. ULI plug-in 58 extends the functionality of browser 50. For example, ULI plug-in
58 extends the functionality of browser 50 to both play multimedia file in a browser
window and allows user to interact with the play multimedia files through the
selection/activation of active areas, as described below.

In one embodiment, ULI plug-in 58 operates as a wrapper for ULI player
52. Thus, ULI player 52, instead of operating in a separate application memory space,
shares the same application memory space as ULI plug-in 58. The operation of ULI
player remains the same regardless of whether ULI plug-in 58 or another application
initiates operation of ULI player 52.

Figure 4 is a flow diagram of one mode of operation of ULI player 52. In
block 150, ULI player 52 is called to initiate the playback of a multimedia file which
corresponds to the selection of the user in browser application 50. For example, the user
could have chosen a selection in browser application 50 to request more information about a particular cruise, vacation or cruise ship. This selection is detected by browser application 50 and passed on to ULI player 52. Operation then continues with block 152.

In block 152, ULI player 52 is loaded into memory 24 and is initialized to begin execution. Operation will then continue with block 154.

In block 154, ULI player 52 will perform the VOLUME function to seek the CD-ROM with the correct label. If there is no CD-ROM with the correct label contained in data storage device 36, the user will be prompted with a dialog screen requesting that the user place the CD-ROM with the correct VOLUME inside data storage device 36. After a CD-ROM with the correct label has been detected, ULI player 52 will also perform the VERSION function, in order to check that the CD-ROM with the correct label is also the correction version of the CD-ROM. In addition, ULI player 52 will also perform the disk function to check to see if the CD-ROM in data storage device 36 has the correct ID number. If VOLUME, VERSION, and DISK functions execute correctly, ULI player 52 will then execute the OPEN command to open the appropriate multimedia file for display. In addition, ULI player 52 will also execute the PUT_WINDOW_AT function to display a window on display device 28 in which MCI system layer 54 will display the multimedia file. ULI player 52 will then perform the PLAY function, and operation will continue with block 156.

In block 156, ULI player 52 will pass MCI system layer 54 the coordinate of the window ULI player 52 created with the PUT_WINDOW_AT function, and ULI player 52 also sends MCI system layer 54 the name and location of the multimedia file from which MCI system layer 54 will retrieve the necessary multimedia information to be displayed in the above-mentioned window. As noted above, this multimedia information not only includes media data, but can include audio data as well. Moreover, this multimedia information can also consist of sprites and other animation information. After ULI player 52 calls MCI system layer 54 and provides the above information, operation will then continue with block 158.

In block 158, MCI system layer 54 will perform the playback of the multimedia file.

The above-described process for displaying video images in a window overlaying a web page (referred to as Parallel Addressing Video Technology), may also be used to provide Dynamic Integrated Video Overlay (DIVO). In Parallel Addressing Video
Technology, a video image is displayed within a window overlaying another image (e.g., a web page). In DIVO, however, instead of the video being displayed within a window, compositing techniques are employed so that the video is created as a foreground image that is composited with a background image so that the video image appears to form part of the background image (i.e., the video image is integrated as part of the web page, instead of within a window overlaying the web page).

In DIVO, a process is described for combining Parallel Addressing Video Technology and HTML to create seamless interactive Internet movies that can use a variety of effects to present creative design and information over the Internet.

In one example, when a user first views a web page created using the technology, what initially appears is a standard web page with a full-screen view of a theater stage and selectable menu items to the left and right side of the stage. The user then sees what appears and sounds like a person walking behind the curtains and asking someone else, “How do I get on to the web page?” The user sees and continues to hear what sounds like confusion just prior to an actor pulling/tearing through the curtains (appearing to tear through the web page) and climbing out through what the user had thought were standard static web graphics. Once on stage, the actors moves around on the stage and introduces the user to the web site. The user can then select various options to see different scenes from the site including among others, curtains opening to a movie screen that starts to play a movie trailer, or advertisements which expand to take over the full computer screen.

The process allows creative artists and designers an almost unlimited number of television and movie style special effects to incorporate into interactive web pages.

In one embodiment, there are five phases to the creation of DIVO pages:

Phase 1: Creative and Technical Scene Design.

Phase 2: Graphic Design of Static Elements.

Phase 3: Video Production and Post Product Effects Creation.

Phase 4: Web Page Integration of Static and Active Elements.

Phase 5: Deployment of DIVO Pages to the Internet.

**Phase 1: Creative and Technical Scene Design**

During this phase, web designers review and determine what their web pages needs to provide the audience/user to effectively present their message and allow the user
to make use of the page interactively. Creative designers determine how they would like the DIVO page to appear to the user.

By way of example, a website designer might determine what is needed is a web page to allow users to view movie trailers for upcoming films, interviews with celebrities, and to be able to purchase tickets at their local theaters for upcoming films.

For the introductory “Home” page, the designer determines that it would be desirable to present this with a scene from within a theater. The designer wants the users to view the scene as if they were members of the audience sitting in a movie theater. The designer also wants a host to tear through the curtains in the scene and then introduce the user to the different parts of the web site.

Part of this design is to determine which portion of the scene would need be live video, which would be animated graphics, and which still images would be needed. The following is an example of suitable selections for this purpose:

A) Live video: The center of the stage and any area where the curtains would move as a result of the host tearing through would need to be live. For example, at the beginning of the scene, the actor’s feet would be seen moving just below the curtain with a voice over of a somewhat confused host asking backstage directors how the host is supposed to get on the stage.

B) Animated Images: A set of icons to the right and left of the stage needs to be created to display the user options to take the user to different areas of the web site. For one example, a display board similar to that at the front of a movie theater would have moving lights. This could be achieved using a series of animated graphics.

C) Still Images: The top and far left/right portions of the curtains do not move during the movement on stage. In addition, images of the stage and front row seats would be static during the action on stage, therefore they may be still images as well.

A production list is now assembled and the other phases can begin.

Phase 2: Graphic Design of Static Elements

Still images are designed as graphic interchange format (GIF) and Joint Photographic Experts Group (JPEG) graphics. They are created using standard web graphic design principles and are optimized for size to allow for faster transfer over all networks, including the Internet. Each of the still images are created first, including a place holding image for each of the animated graphics.
Background and overlay images (also still) are created for use in the video, these can be props, backdrops, etc. Animated GIF graphics are then created to replace the place-holder graphics.

**Phase 3: Video Production and Post Production Effects Creation**

**A) Video Shoot**

A video shoot is done on a stage or set using green/blue screens for creating a composite image in the same manner that a weather person or special effects in a movie are done. The stage area floor also receives a green or blue screen treatment to provide total control over the scene. The outfits for actors and place-holder props used in the scene are a different color than the green or blue screen being used in the background (and in some cases, portions of the foreground).

All of the video footage captured in the video shoot is done in a format as close to zero-compression as possible (e.g., Sony BetaSP and Digital Beta). The largest subject shot available is done and camera angles and sets are designed with this principle in mind.

For example: during a video shoot for the above described scenario, the actor is required to stand behind a green screen curtain and then tear/climb through the curtain to end up in front of it. The actor then does an introduction before climbing back through the curtain.

**B) Image Creation**

Background images other than the first frame images (created in Phase 2) are created for the scene. Graphics for any of the place-holder props used during the shoot are also created. These are created as high-resolution (TIF) graphics.

**C) Digital Capture**

The video footage is processed using a digital capture board attached to a computer. This footage is captured as close to zero-compression as possible using advanced equipment (e.g., a Targa 2000 RTX capture peripheral). A high quality video codec is used to format the video into a computer readable digital video at zero-compression (e.g., a Targa Quicktime Codec).

**D) Basic Editing**

Basic editing is done using digital video software tools (e.g., Adobe Premiere video editing software). During this edit phase, standard video cuts are performed and there is piecing together of various shots (e.g., removing actors mistakes, adding tape
segments, etc.). Audio fixes and level corrections are also done at this point. The edited version is then output to a zero-compression format (e.g., Apple Computer, Inc.'s QuickTime format).

E) Effects Editing

The video is then imported into a video effects software (e.g., Adobe After Effects and various plug-ins for that application). Further editing is done and special effects are added at this point. Compositing is done using a chroma-key to replace the green/blue screen with our background images and prop place-holder images according to well known techniques. (e.g., a Boris FX Chroma Key Plug-In) The file is then output to an uncompressed master video.

F) Compression to MPEG Standard

The video is then compressed into two different formats following the Motion Picture Experts Group (MPEG) format. Format 1 is compressed to a high resolution of 384 x 288 pixels with a data rate of 360 kb/second. Standard video is compressed to a resolution of 320 x 240 pixels at a rate of 160 kb/second for non-DIVO, and compressed at 320 x 240 pixels at a rate of 360 kb/second for DIVO format. The different video formats are created to allow maximum usability with there being a variant level of support in the MPEG standard between various vendors of computer hardware and software. Optionally, the video may be encrypted during the processing of the video to prevent unauthorized access to the content of the multimedia file. Access may be then provided if a user provides the proper access codes or passwords, as described below.

Phase 4: Web Page Integration of Static and Active Elements

At this point, the web page may incorporate both the video and the still and animated graphic elements created in the other phases of development. The video portion of the screen starts as a single-image (e.g., using the first frame of video), overlaid on top of a still background image to provide what appears to be a standard web page.

ULI player 52 and ULI plug-in 58 are used to provide the enhanced functionality of DIVO as follows:

A) ULI player 52 and ULI plug-in 58 are a web browser “helper application” and a web browser “plug-in”, respectively. Thus, ULI plug-in 58 operates as a browser plug-in component for browser application 50 and interfaces with ULI player 52.

B) The ULI player 52 is, in one preferred embodiment, a 32-bit application.
C) The use of ULI plug-in 58 allows the integration of video within a browser window. This enhanced capability allows the embedding of video within any selected area of the browser client window. It also maintains compatibility with existing HTML code. Full screen video can be displayed by specifying the full-size of the browser window, or by launching the separate “helper application”, ULI player 52, to display it in a separate window.

D) In one embodiment, ULI plug-in 58 uses the Netscape Corporation Plug-In Application Programming Interface (API), which can also be called through Microsoft Corporation’s Internet Explorer web browser software.

E) ULI plug-in 58 also has Vector-Image-Mapping (VIM) functionality allowing videos to become interactive. Hyperlink areas within each video are mapped out with a set of co-ordinates. Several areas can be mapped out, each can be selectable for a set of frames. By selecting one of the hyperlink areas during video playback, the user can move within the video to another section, link to a specified URL, or even call an external program.

F) As described above, video and other multimedia information is distributed on a CD-ROM, DVD, or a DVD-ROM. In addition, a VIM file is distributed with each video produced, either on the same distribution media as the distributed multimedia, or separately. This file contains the name of the actual video file to be played, and the various commands to select areas of the video as hyperlinks. Thus, the file contains the list of areas within each video that are VIM enabled and the respective hyperlinks for these areas. Table 1 contains a listing of the fields used in the VIM file. In addition, MCI system layer 54 is modified to use a media interpreter/player such as DirectShow by Microsoft Corporation.

ULI plug-in 58 is called by the browser in place of the “helper-application”, ULI player 52. A VIM program has been created to address the end-user needs in accessing hyperlinks within a video.

**VIM File Structure**

The VIM file structure is a collection of variable length chunks. All chunks consist of a chunk header and a body. Before all else is the file signature with is 3 bytes long “VIM”: 
**Table 1**

### Chunk Header:
- **Chunk ID**: 4 bytes (int)
- **Chunk Length**: 4 bytes (int)

<table>
<thead>
<tr>
<th>Chunks</th>
<th>Chunk Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIM version</td>
<td>4 bytes (float)</td>
<td>VIM file version</td>
</tr>
<tr>
<td>Media path</td>
<td>String</td>
<td>Path the VIM media file – not used by the player</td>
</tr>
<tr>
<td>Event count</td>
<td>4 bytes (int)</td>
<td>Number of events in to be triggered</td>
</tr>
<tr>
<td>Event</td>
<td>variable</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
<td>Name of event</td>
</tr>
<tr>
<td>Frame</td>
<td>4 bytes (int)</td>
<td>Frame to trigger event on</td>
</tr>
<tr>
<td>Action</td>
<td>4 bytes (int)</td>
<td>Action to take for this event</td>
</tr>
<tr>
<td>Pause</td>
<td>4 bytes (int)</td>
<td>Pause the media if event triggered</td>
</tr>
<tr>
<td>Go to frame</td>
<td>4 bytes (int)</td>
<td>Go to frame if event triggered</td>
</tr>
<tr>
<td>Play sound</td>
<td>4 bytes (int)</td>
<td>Play sound if event triggered</td>
</tr>
<tr>
<td>Source</td>
<td>4 bytes (int)</td>
<td>Used by action (Example: <a href="http://www.hello.com">http://www.hello.com</a>)</td>
</tr>
<tr>
<td>Object count</td>
<td>4 bytes (int)</td>
<td>Number of objects in this VIM file</td>
</tr>
<tr>
<td>Object</td>
<td>variable</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
<td>Name of object</td>
</tr>
<tr>
<td>Keyframe count</td>
<td>4 bytes (int)</td>
<td>Number of keyframes in this VIM file</td>
</tr>
<tr>
<td>Frame</td>
<td>4 bytes (int)</td>
<td>Keyframe name</td>
</tr>
<tr>
<td>Window Rect</td>
<td>16 bytes (int x 4)</td>
<td>Keyframe window coordinates</td>
</tr>
<tr>
<td>Cursor type</td>
<td>4 bytes (int)</td>
<td>Type of cursor to show when over the keyframe</td>
</tr>
<tr>
<td>Key trigger</td>
<td>4 bytes (int)</td>
<td>Keyboard key that triggers this keyframe</td>
</tr>
<tr>
<td>Mouse trigger</td>
<td>4 bytes (int)</td>
<td>Mouse button that triggers this keyframe</td>
</tr>
<tr>
<td>Show type</td>
<td>4 bytes (int)</td>
<td>Show keyframe when (Always, Never, Cursor)</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
<td>Text to show for this keyframe</td>
</tr>
<tr>
<td>Action</td>
<td>4 bytes (int)</td>
<td>Action to be taken if keyframe triggered</td>
</tr>
<tr>
<td>Pause</td>
<td>4 bytes (int)</td>
<td>Pause the media if keyframe triggered</td>
</tr>
<tr>
<td>Go to frame</td>
<td>4 bytes (int)</td>
<td>Go to frame if keyframe triggered</td>
</tr>
<tr>
<td>Play sound</td>
<td>4 bytes (int)</td>
<td>Play sound if keyframe triggered</td>
</tr>
<tr>
<td>Source</td>
<td>4 bytes (int)</td>
<td>Used by action (Example: <a href="http://www.hello.com">http://www.hello.com</a>)</td>
</tr>
<tr>
<td>End of file</td>
<td>0 bytes</td>
<td></td>
</tr>
</tbody>
</table>

**ULI File Structure**

The ULI file structure is a collection of variable length chunks. All chunks consist of a chunk header and a body. Before all else in the file signature with is 3 bytes long “ULI”:

5
Phase 5: Deployment of DIVO Pages to the Internet.

Deployment can be accomplished as follows:

1) Web documents are prepared for the web site, including the creation of the video files.

2) Video files are played in the VIM editor. Selections and their corresponding hyper-links are made. These are saved to a VIM file.

3) Video files and VIM files are stored on media (CD-Rom, DVD, DVD-ROM).

4) Settings for active areas are adjusted in ULI plug-in 58 rather than in the web browser, and the active areas are then displayed on the users screen, as described below.

Standard page design principles are used throughout for the page layout, and by using the Parallel Addressing Video Technology described above, it is possible to integrate the video seamlessly as part of the scene comprised of still images and animated

---

**Table 2**

<table>
<thead>
<tr>
<th>Chunks</th>
<th>Chunk Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ULI version</td>
<td>4 bytes (float)</td>
<td>ULI file version</td>
</tr>
<tr>
<td>ULI license</td>
<td>string</td>
<td>Ownership of this ULI file</td>
</tr>
<tr>
<td>ULI expiration</td>
<td>8 bytes (double)</td>
<td>ULI file expiration date</td>
</tr>
<tr>
<td>Volume count</td>
<td>4 bytes (int)</td>
<td>Number of volume names in this ULI file</td>
</tr>
<tr>
<td>Volume name</td>
<td>string</td>
<td>The volume name itself</td>
</tr>
<tr>
<td>next name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Volume end</td>
<td>0 bytes</td>
<td>End of volume names</td>
</tr>
<tr>
<td>Volume ID</td>
<td>String</td>
<td>Compare this ID with the one on the CD</td>
</tr>
<tr>
<td>Volume version</td>
<td>4 bytes (int)</td>
<td>Compare this version with the one on the CD</td>
</tr>
<tr>
<td>Window caption</td>
<td>string</td>
<td>ULI player window caption</td>
</tr>
<tr>
<td>URL redirection</td>
<td>string</td>
<td>Go to this web page upon exit</td>
</tr>
<tr>
<td>Media type</td>
<td>4 bytes (int)</td>
<td>Type of media (MPEG, AUDIO, VIM)</td>
</tr>
<tr>
<td>Media path</td>
<td>String</td>
<td>Path to the media file</td>
</tr>
<tr>
<td>Start offset</td>
<td>4 bytes (int)</td>
<td>Start the media at this frame</td>
</tr>
<tr>
<td>Auto start</td>
<td>4 bytes (int)</td>
<td>Auto play the media</td>
</tr>
<tr>
<td>Media loop</td>
<td>4 bytes (int)</td>
<td>Continuously loop the media from start to finish</td>
</tr>
<tr>
<td>Controls on</td>
<td>4 bytes (int)</td>
<td>Display the media controls in the window</td>
</tr>
<tr>
<td>Controls allowed</td>
<td>4 bytes (int)</td>
<td>Allow the media controls to be activated</td>
</tr>
<tr>
<td>Full screen</td>
<td>4 bytes (int)</td>
<td>Start the player in full screen mode</td>
</tr>
<tr>
<td>License code</td>
<td>4 bytes (int)</td>
<td>Encryption code to prevent editing of ULI file</td>
</tr>
<tr>
<td>End of file</td>
<td>0 bytes</td>
<td></td>
</tr>
</tbody>
</table>
graphics. As a result, the total page comes to life as an interactive video without the need for very large video formats (full-screen video), and without the user recognizing that the video is only a small part of the overall screen. The Parallel Addressing Video Technology, combined with HTML, displays the MPEG video at a rate of 30 frames per second, or, close to television quality.

Figure 5 is a flow diagram illustrating the operation of the system to offer DIVO functionality in accordance with one embodiment of the present invention. The process starts when, after a user has opened a browser application, it is connected with a web browser containing DIVO pages.

In block 500, the user initiates a download of an ULI file (i.e., a file with a "ULE" file extension). The user may initiate the download of the ULI file through the browser application's menu system, or by clicking on a link on the displayed web page in the browser application. After the user initiates a download of the ULI file, browser application 50 initializes ULI player 52 through the use of ULI plug-in 58. In one embodiment, after browser application detects the completed download of the ULI file, browser application 50 determines the application that is assigned to handle the processing of the ULI file. Thus, browser application 50 associates the ULI file with ULI plug-in 58 and pass the ULI file reference to ULI plug-in 58. ULI plug-in 58 then initiates ULI player 52 and run ULI player 52 with the ULI file. Operation will then continues with block 502.

In block 502, the file command interpreter of ULI player 52 seeks for and loads the VIM file based on the content of the ULI file. In one embodiment, the ULI file contains the location of the VIM file and the multimedia file (e.g., video file) that is associated with the VIM file. As discussed above, each VIM file has an associated multimedia file. The file command interpreter of ULI player 52 optionally retrieves a key to decrypt any information in the multimedia file. In another embodiment, the ULI file contains location information for an authentication server from which a key may be retrieved once the user has provided proper authentication. The authentication process could involve the authentication server asking the user for a user name/password and then provide a decryption key to ULI player 52. After the file command interpreter loads the VIM file, operation continues with block 504. In block 504, ULI player 52 parses the contents of the VIM file, locate all segments of the multimedia files which are to be
played, and sets up active areas (e.g., “hot spots”), for a respective segment of multimedia. Operation then continues with block 506.

In block 506, ULI player 52 plays the multimedia sequence (e.g., video/audio media) sequence, and, through the functionality provided by plug-in 58, the multimedia sequence appears integrated into the current background. In the embodiment where the multimedia file is encrypted, ULI player 52 decrypts the multimedia file with the key that is retrieved in block 502. As ULI player 52 is playing the multimedia sequence, ULI player 52 also determines if the user has selected an active area in the area where the multimedia sequence is being presented. In other embodiments, the active areas may offer a rollover-type of interactivity. For example, if a user puts a cursor above an active area in the multimedia window, but does not click the mouse button, certain actions may be triggered simply by the movement of cursor over that area.

In block 508, ULI player 52 detects whether an active area has been triggered by the user. If an active area has been triggered by the user, then operation continues with block 512 where ULI player 52 processes the selection event. If an active area has not been triggered by the user, then operation continues with block 510, where ULI player 52 continues with block 510.

In block 510, ULI player 52 detects if there are additional segments of multimedia files to be played. If so, operation continues with block 504, where the next segment is loaded and the corresponding active areas are configured. If there are no additional segments of multimedia files to be played, ULI player 52 stops seeking additional segments.

In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.
CLAIMS

What is claimed is:

1. A method for presenting a video image and compositing the video image
   with a static web page in a browser application comprising:
   a) receiving a request from a user to initiate the playback of the video image
   stored on a pre-distributed medium;
   b) transferring, from a network, a control file containing location information
   for the video image and a vector-image-mapping file associated with the video image;
   c) processing the vector-image-mapping file to retrieve a set of active areas
   coordinates;
   d) associating the set of active areas coordinates with a set of areas in the
   video image; and,
   e) presenting the video image while determining user interactivity within the
   set of areas in the video image.

2. The method of claim 1, further comprising loading the vector-image-
   mapping file and the video image from the pre-distributed medium.

3. The method of claim 1, further comprising activating a link based on the
   user interactivity with the set of active areas.

4. The method of claim 1, wherein the video image is encrypted and
   presenting the video image includes decrypting the video image.

5. The method of claim 4, wherein the control file contains a decryption key
   and decrypting the video image includes retrieving the decryption key from the control
   file.

6. The method of claim 1, wherein determining user interactivity within the
   set of areas in the video image includes determining if the user has issued a selection
   command within the set of areas.
7. The method of claim 6, wherein determining if the user has issued a
selection command within the set of areas includes determining if the user has issued the
selection command with a cursor control device while a cursor is within the set of areas.

8. The method of claim 6, wherein determining if the user has issued a
selection command within the set of areas includes determining if the user has issued the
selection command with an input device while a graphical selection indicator is within the
set of areas.

9. An article comprising a computer readable medium having instructions
stored thereon, which when executed, causes:
   a) receipt of a request from a user to initiate the playback of the video image
      stored on a pre-distributed medium;
   b) transfer of a control file from a network containing location information
      for the video image and a vector-image-mapping file associated with the video image;
   c) process of the vector-image-mapping file to retrieve a set of active areas
      coordinates;
   d) association of the set of active areas coordinates with a set of areas in the
      video image; and,
   e) presentation of the video image while determining user interactivity within
      the set of areas in the video image.

10. The article of claim 7, where the computer readable medium having
    additional instructions stored thereon, which when executed, causes loading the vector-
    image-mapping file and the video image from the pre-distributed medium.

11. The article of claim 7, where the computer readable medium having
    additional instructions stored thereon, which when executed, causes activating a link
    based on the user interactivity with the set of active areas.

12. The article of claim 7, wherein the video image is encrypted and
    presenting the video image includes decrypting the video image.
13. The article of claim 12, wherein the control file contains a decryption key and decrypting the video image includes retrieving the decryption key from the control file.

14. The article of claim 12, wherein the control file contains location information for a security server and decrypting the video image includes retrieving a decryption key from the security server.

15. The article of claim 14, wherein retrieving a decryption key from the security server includes receiving a user name and a password from a user; querying the secure server with the user name and the password; and returning the decryption key if the user name and the password is validated.

16. The article of claim 10, wherein determining user interactivity within the set of areas in the video image includes determining if the user has issued a selection command within the set of areas.

17. The article of claim 16, wherein determining if the user has issued a selection command within the set of areas includes determining if the user has issued the selection command with a cursor control device while a cursor is within the set of areas.

18. The article of claim 16, wherein determining if the user has issued a selection command within the set of areas includes determining if the user has issued the selection command with an input device while a graphical selection indicator is within the set of areas.

19. An apparatus comprising:
   a) receiving a request from a user to initiate the playback of the video image stored on a pre-distributed medium;
   b) transferring a control file from a network containing location information for the video image and a vector-image-mapping file associated with the video image;
   c) processing the vector-image-mapping file to retrieve a set of active areas coordinates;
d) associating the set of active areas coordinates with a set of areas in the video image; and,
e) presenting the video image while determining user interactivity within the set of areas in the video image.

20. The apparatus of claim 19, further comprising means for loading the vector-image-mapping file and the video image from the pre-distributed medium.

21. The apparatus of claim 19, further comprising means for activating a link based on the user interactivity with the set of active areas.

22. The apparatus of claim 19, wherein the video image is encrypted and the means for presenting the video image includes means for decrypting the video image.

23. The apparatus of claim 22, wherein the control file contains a decryption key and the means for decrypting the video image includes means for retrieving the decryption key from the control file.

24. The apparatus of claim 22, wherein the control file contains location information for a security server and the means for decrypting the video image includes means for retrieving a decryption key from the security server.

25. The apparatus of claim 24, wherein the means for retrieving a decryption key from the security server includes means for receiving a user name and a password from a user; means for querying the secure server with the user name and the password; and means for returning the decryption key if the user name and the password is validated.

26. The apparatus of claim 19, wherein means for determining user interactivity within the set of areas in the video image includes means for determining if the user has issued a selection command within the set of areas.

27. The apparatus of claim 19, wherein the means for determining if the user has issued a selection command within the set of areas includes means for determining if the user has issued the selection command with a cursor control device while a cursor is within the set of areas.
28. The apparatus of claim 19, wherein the means for determining if the user has issued a selection command within the set of areas includes means for determining if the user has issued the selection command with a input device while a graphical selection indicator is within the set of areas.
START

Browser Application Calls ULPlayer

Initialize ULPlayer

File Command Interpreter and Search for CD-ROM

Passes a Sequence Call into Windows Media Command Interrupter (MCI) System Layer

Video/Audio Media is Played in Windows

END

Figure 4
START

Browser Application Initializes UL Plug-In After User Initiates Download Of ULI File (.ule)

File Command Interpreter Loads VIM File (.vim) Based on ULI File

Parse VIM, Locate Segments, and Set-up Active Areas (e.g., "Hot Spots")

Play Video/Audio Media Sequence

Active Areas Triggered?

YES → Process Selection Event

NO → Additional Segments?

YES → ...

NO → END

Figure 5
# INTERNATIONAL SEARCH REPORT

**International application No.**

PCT/US00/09779

## A. CLASSIFICATION OF SUBJECT MATTER

<table>
<thead>
<tr>
<th>IPC(7)</th>
<th>H04N 7/10; H04M 11/00; G06F 13/00</th>
</tr>
</thead>
<tbody>
<tr>
<td>US CL</td>
<td>348/7; 395/200.02; 395/154</td>
</tr>
</tbody>
</table>

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

| U.S.          | 348/7; 395/200.02; 395/154 |

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

IEEE, ACM

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

West, East, STN

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US 5,568,181 A (GREENWOOD et al.) 22 October 1996, col 1, lines 33 - col 8, lines 8</td>
<td>1-28</td>
</tr>
<tr>
<td>Y</td>
<td>US 5,805,804 (LAURSEN et al.) 08 September 1998, col 2, lines 14 - col 24, lines 67</td>
<td>1-28</td>
</tr>
<tr>
<td>Y</td>
<td>US 5,564,001 A (LEWIS) 08 October 1996, col 2, lines 58 - col 15, lines 67</td>
<td>1-28</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. | See patent family annex.

- Special categories of cited documents
  - "A" document defining the general state of the art which is not considered to be of particular relevance
  - "E" earlier document published on or after the international filing date
  - "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  - "O" document referring to an oral disclosure, use, exhibition or other means
  - "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

- "X" document of particular relevance, the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

- "Y" document of particular relevance, the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

- "&" document member of the same patent family

### Date of the actual completion of the international search

18 JUNE 2000

### Date of mailing of the international search report

19 JUL 2000

Name and mailing address of the ISA/US

Commissioner of Patents and Trademarks

Box PCT

Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer: [Signature]

Alvin Oberley

Telephone No. (703) 305-9716

Form PCT/ISA/210 (second sheet) (July 1998)