Technical Disclosure Commons

Defensive Publications Series

December 15, 2017

Cylindrical User Interface with Infinite Scrolling

Michael Alger

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Alger, Michael, "Cylindrical User Interface with Infinite Scrolling", Technical Disclosure Commons, (December 15, 2017)
http://www.tdcommons.org/dpubs_series/1016

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Cylindrical User Interface with Infinite Scrolling

Abstract:

Techniques are described that enable a cylindrical user interface with infinite scrolling. The cylindrical user interface enables a user to scroll through content that is arranged in a self-refreshing loop around the user. The user may remain stationary and rotate the cylinder to view and select content. As the content circles the user, already-viewed content is temporarily destroyed behind the user and new content is provided, allowing the user to scroll in one direction and have new content continuously presented. In other implementation, the user can turn around in the center of the cylinder and as the user makes a complete rotation, the interface will temporarily destroy previously viewed content and replace it with new content.

Keywords:

Virtual reality, user interface, cylindrical user interface, scrolling, infinite scrolling, cylindrical scrolling, virtual reality controls

Background:

Virtual reality (VR) environments rely on display, tracking, and VR-content systems. Through these systems, realistic images, sounds, and sometimes other sensations simulate a user's physical presence in an artificial environment. Each of these three systems are illustrated below in Fig. 1.

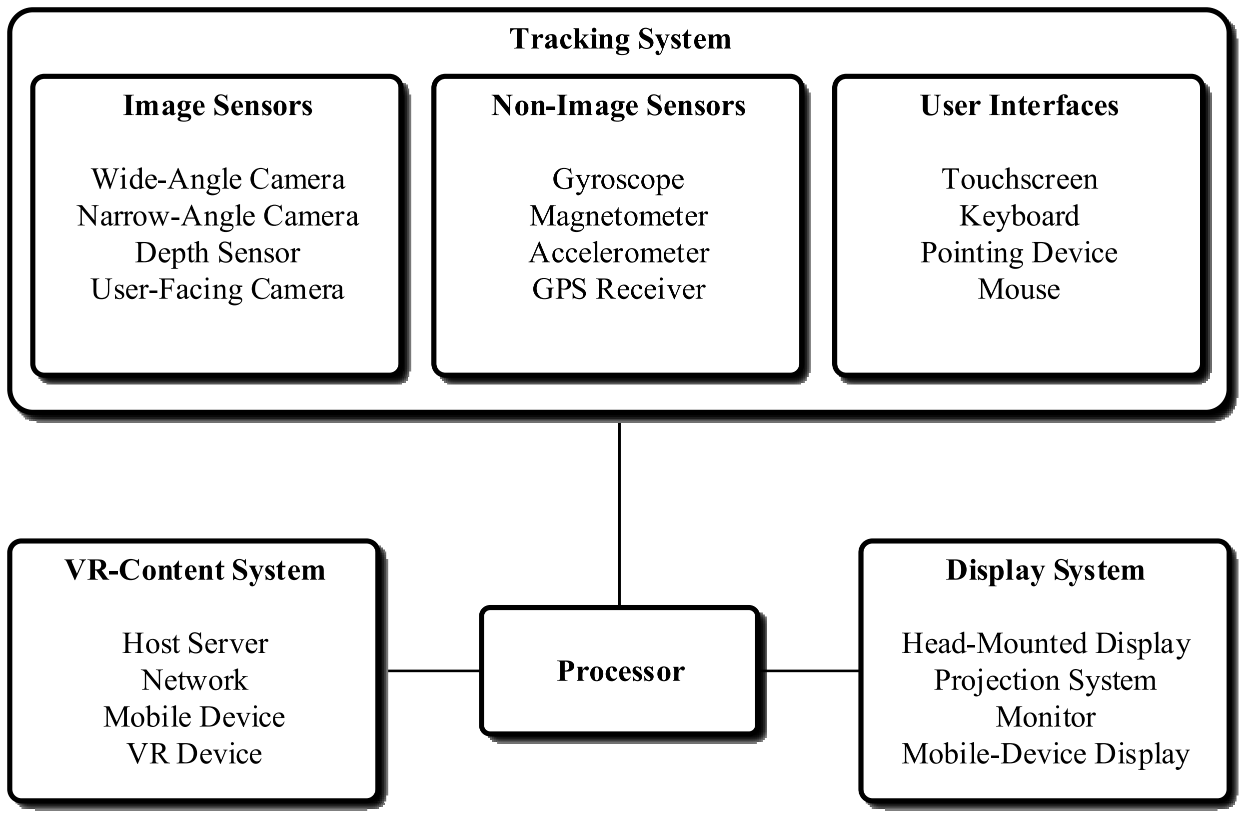


Fig. 1

The systems described in Fig. 1 may be implemented in one or more of various computing devices that can support VR applications, such as servers, desktop computers, VR goggles, computing spectacles, laptops, or mobile devices. These devices include a processor that can manage, control, and coordinate operations of the display, tracking, and VR-content systems. The devices also include memory and interfaces. These interfaces connect the memory with the systems using various buses and other connection methods as appropriate.

The display system enables a user to "look around" within the virtual world. The display system can include a head-mounted display, a projection system within a virtual-reality room, a monitor, or a mobile device's display, either held by a user or placed in a head-mounted device.

The VR-content system provides content that defines the VR environment, such as images and sounds. The VR-content system provides the content using a host server, a network-based device, a mobile device, or a dedicated virtual reality device, to name a few.

The tracking system enables the user to interact with and navigate through the VR environment, using sensors and user interfaces. The sensors may include image sensors such as a wide-angle camera, a narrow-angle camera, a user-facing camera, and a depth sensor. Non-image sensors may also be used, including gyroscopes, magnetometers, accelerometers, GPS sensors, retina/pupil detectors, pressure sensors, biometric sensors, temperature sensors, humidity sensors, optical or radio-frequency sensors that track the user's location or movement (*e.g.*, user's fingers, arms, or body), and ambient light sensors. The sensors can be used to create and maintain virtual environments, integrate "real world" features into the virtual environment, properly orient virtual objects (including those that represent real objects, such as a mouse or pointing device) in the virtual environment, and account for the user's body position and motion.

The user interfaces may be integrated with or connected to the computing device and enable the user to interact with the VR environment. The user interfaces may include a touchscreen, a keyboard, a pointing device, a mouse or trackball device, a joystick or other game controller, a camera, a microphone, or an audio device with user controls. The user interfaces allow a user to interact with the virtual environment by performing an action, which causes a corresponding action in the VR environment (*e.g.,* raising an arm, walking, or speaking).

The tracking system may also include output devices that provide visual, audio, or tactile feedback to the user (*e.g.*, vibration motors or coils, piezoelectric devices, electrostatic devices, LEDs, strobes, and speakers). For example, output devices may provide feedback in the form of blinking and/or flashing lights or strobes, audible alarms or other sounds, songs or other audio files, increased or decreased resistance of a control on a user interface device, or vibration of a physical component, such as a head-mounted display, a pointing device, or another user interface device.

Fig. 1 illustrates the display, tracking, and VR-content systems as disparate entities in part to show the communications between them, though they may be integrated, *e.g.*, a smartphone mounted in VR goggles, or operate separately in communication with other systems. These communications can be internal, wireless, or wired. Through these illustrated systems, a user can be immersed in a VR environment. While these illustrated systems are described in the VR context, they can be used, in whole or in part, to augment the physical world. This augmentation, called "augmented reality" or AR, includes audio, video, or images that overlay or are presented in combination with the real world or images of the real world. Examples include visual or audio overlays to computing spectacles (*e.g.,* some real world-VR world video games or information overlays to a real-time image on a mobile device) or an automobile's windshield (*e.g.,* a heads-up display) to name just a few possibilities.

A particular configuration of the systems of Fig. 1 presents the user with a cylindrical user interface with infinite scrolling that allows the user to scroll through content, such as images or a game library, arranged in an infinite loop around the user. For example, the user may remain stationary and rotate the cylinder to find an image, game, or application to select. As the content circles the user, it is temporarily destroyed and new content is provided so that the user can scroll in one direction and have new content continuously presented. In other implementation, the user can turn around in the center of the cylinder and, as the user makes a complete rotation, the interface will temporarily destroy previously viewed content and replace it with new content.

An example of a cylindrical user interface with infinite scrolling is illustrated in Fig. 2, which shows a user standing in the cylinder with six panels representing six content items. As shown, the user can grab the panels and rotate them around to see the other panels or turn around to see the other panels. In both implementations, the user sees new content as long as the rotation continues in the same direction. If the user turns the other direction or scrolls the other direction, previously viewed content can be brought back in reverse order.

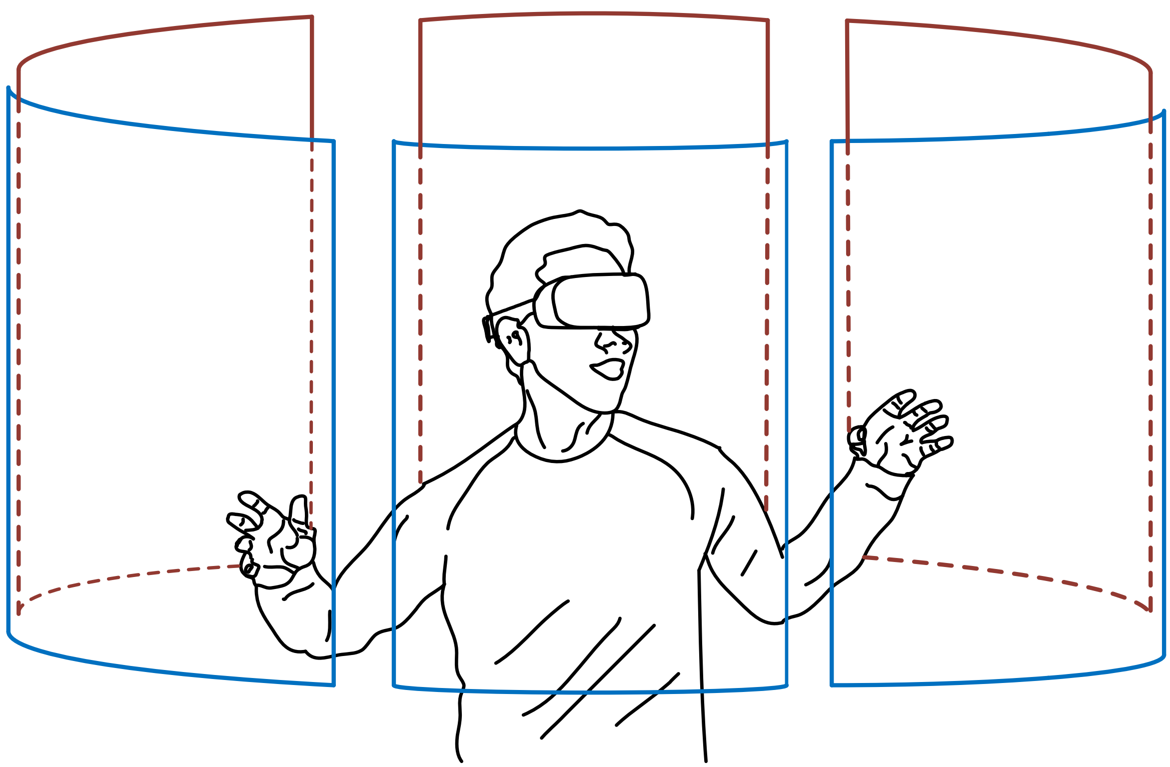


Fig. 2

Description:

Techniques are described that enable a cylindrical user interface with infinite scrolling. The cylindrical user interface enables a user to scroll through content that is arranged in a self-refreshing loop around the user. The user may remain stationary and rotate the cylinder to view and select content. As the content circles the user, already-viewed content is temporarily destroyed behind the user and new content is provided, allowing the user to scroll in one direction and have new content continuously presented. In other implementation, the user can turn around in the center of the cylinder and as the user makes a complete rotation, the interface will temporarily destroy previously viewed content and replace it with new content.

Fig. 3 illustrates four top-views (3-1, 3-2, 3-3, and 3-4) of the user interacting with an example implementation of the cylindrical user interface with infinite scrolling. In the example implementation shown in Fig. 3, the user is wearing a head-mounted display and scrolling the user interface in a counter-clockwise direction, as shown by the arrows. Five groups of colored lines are used in Fig. 3 to illustrate destroyed and new content (orange, violet, dark blue, light blue, and turquoise). For clarity, the groups of colored lines are also separated by tick marks. The user's general viewpoint is shown by the blue wedge extending from the head-mounted display. In view 3-1, five panels of content are visible. Four panels are shown as orange lines and one as a violet line. In view 3-2, the user has scrolled the four orange panels counter-clockwise and three new violet panels have been provided (for a total of four violet panels).

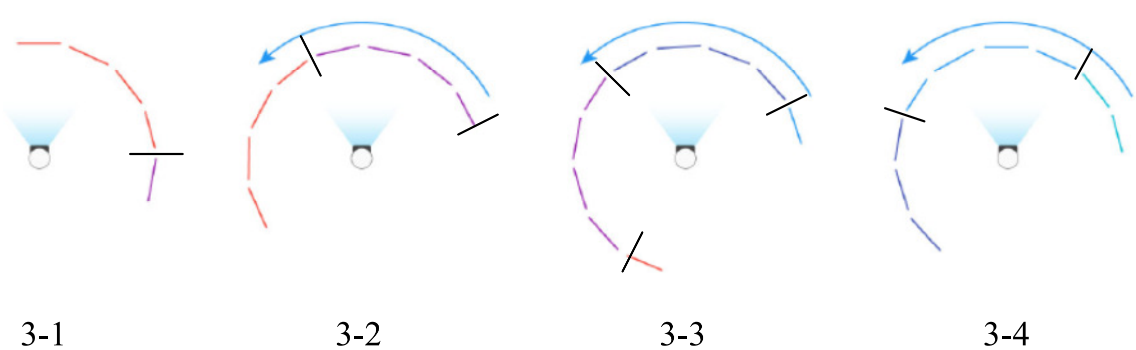
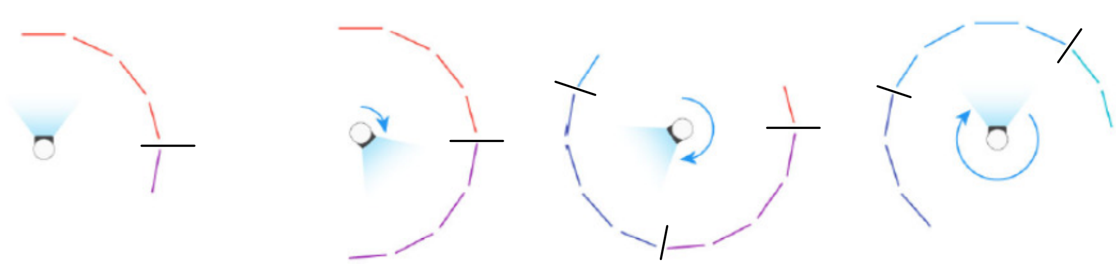


Fig. 3

View 3-3 shows that as the user continues to scroll counter-clockwise, content is temporarily destroyed and new content is created. Thus, three of the orange lines are gone, the four violet lines remain, and five new lines have been created, four dark blue lines and one light blue line. In view 3-4, the user continues counter-clockwise scrolling, and all of the orange and violet lines, along with one of the dark blue lines, are gone. In their place, three more light blue lines have been created (for a total of four), as well as two new turquoise lines. While not shown in Fig. 3, the user may scroll in the opposite direction (clockwise) and the panels will be destroyed and created in the reverse order, allowing the user to return to the panels shown in view 3-1. In some cases, the user may then continue to scroll clockwise and view new content. In other cases, view 3-1 shows a default start position and additional clockwise scrolling is not available.

Fig. 4 illustrates four top-views (4-1, 4-2, 4-3, and 4-4) of the user interacting with another example implementation of the cylindrical user interface with infinite scrolling. In the example implementation shown in Fig. 4, the user is wearing a head-mounted display and turning within the user interface in a clockwise direction, as shown by the arrows. The same five groups of colored lines (orange, violet, dark blue, light blue, and turquoise) and tick marks are used in Fig. 4 to illustrate destroyed and new content. In view 4-1, five lines (panels of content) are visible, four orange lines and one violet line.



4-1 4-2 4-3 4-4

Fig. 4

In view 4-2, the user has turned clockwise, and three new violet panels have been provided (for a total of four violet lines). The four orange lines are still visible. View 4-3 shows that as the user continues to turn in a clockwise direction, previously viewed content is temporarily destroyed and new content is created. Thus, three of the orange lines are gone, the four violet lines remain, and five new lines have been created, four dark blue lines and one light blue line. In view 4-4, the user continues turning clockwise, and all of the orange and violet lines, along with one of the dark blue lines, are gone. In their place, three more light blue lines have been created (for a total of four), as well as two new turquoise lines. As with the example implementation described with reference to Fig. 3, the user may turn in the opposite direction (counter-clockwise) and the panels will be destroyed and created in the reverse order, allowing the user to return to the panels shown in view 4-1. In some implementations, the user may then continue to scroll clockwise and view new content. In other implementations, view 4-1 shows a default start position and additional counter-clockwise turning does not present new content.

The cylindrical user interface with infinite scrolling may use a variety of techniques to determine when to destroy and create new content. In some cases, the VR system (*e.g.*, the tracking system of Fig. 1) may be able to determine the user's head orientation and present the cylindrical user interface based on the head position. In other implementations, the cylindrical user interface can be based on a default position that can be preset, or calibrated by the user the first time the cylindrical user interface is activated.