Systems and methods of camera-based fingertip tracking

Andrew Sugaya

University of Central Florida

Find similar works at: https://stars.library.ucf.edu/patents

University of Central Florida Libraries http://library.ucf.edu

This Patent is brought to you for free and open access by the Technology Transfer at STARS. It has been accepted for inclusion in UCF Patents by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

Recommended Citation


https://stars.library.ucf.edu/patents/579
Systems and methods of camera-based fingertip tracking are disclosed. One such method includes identifying at least one location of a fingertip in at least one of the video frames, and mapping the location to a user input based on the location of the fingertip relative to a virtual user input device.

13 Claims, 4 Drawing Sheets
FIG. 3

FIG. 5
FIG. 4

1. Identify target pixels corresponding to potential fingertips (410).
2. Track motion of target pixels (420).
3. Determine whether target pixels are confirmed pixels (430).
4. Track motion of confirmed pixels corresponding to fingertips (440).
5. Map fingertip position(s) and/or motion to user input (450).
6. Direct user input to application (460).
FIG. 6

APPLICATION 680

UI EVENT 660

OPERATING SYSTEM 610

CAMERA DRIVER 690

MOUSE EVENT 640

KBD EVENT 650

FINGERTIP TRACKING LOGIC 250

APP-SPECIFIC MAP 695

MOUSE DRIVER 620

KEYBOARD DRIVER 630

FIG. 7

APPLICATION 680

UI EVENT 660

OPERATING SYSTEM 610

MOUSE EVENT 640

KBD EVENT 650

CAMERA DRIVER 690

MOUSE DRIVER INTER. 710

KBD DRIVER INTER. 720

APP-SPECIFIC MAP 695

FINGERTIP TRACKING LOGIC 250
1
SYSTEMS AND METHODS OF
CAMERA-BASED FINGERTIP TRACKING

CROSS REFERENCE TO RELATED
APPLICATIONS

This application claims priority to U.S. Ser. No. 61/015, 243, filed Dec. 20, 2007, which is entirely incorporated herein by reference.

TECHNICAL FIELD

The present disclosure relates to visual tracking of objects, and more specifically, to camera-based tracking of fingertips.

BACKGROUND

Conventional mechanisms for user input to computers include keyboards and pointing devices. Touch-sensing displays are also used, but these displays are expensive, and a large size touch-sensing display requires more exertion from the user to perform the same action. Furthermore, a touch-sensing display allows fingertip movement only on a flat surface, and cannot recognize finger tip movement in free space. Fingertip tracking in free space has been accomplished by the use of infrared, and through combination of lasers and steering mirrors. But these methods require complicated, expensive equipment and prove to be impractical. Methods for tracking fingers in free space with ordinary cameras have been developed, but fingers have less mobility and dexterity as compared to fingertips.

BRIEF DESCRIPTION OF THE DRAWINGS

Many aspects of the disclosure can be better understood with reference to the following drawings. The components in the drawings are not necessarily to scale, emphasis instead being placed upon clearly illustrating the principles of the present disclosure.

FIG. 1 depicts one embodiment of a system for camera-based fingertip tracking.

FIG. 2 is a hardware block diagram of one embodiment of the computer from FIG. 1.

FIG. 3 illustrates a pixel grid utilized by the fingertip tracking logic from FIG. 2.

FIG. 4 is a flowchart of the operation in one embodiment of the fingertip tracking logic from FIG. 2.

FIG. 5 illustrates the operation of the mapping block from FIG. 4.

FIG. 6 depicts one example architecture which can be used to implement the fingertip tracking logic from FIG. 2.

FIG. 7 depicts another example architecture which can be used to implement the fingertip tracking logic from FIG. 2.

SUMMARY

Systems and methods for camera-based fingertip tracking are disclosed. One such method includes identifying at least one location of a fingertip in at least one of the video frames, and mapping the location to a user input based on the location of the fingertip relative to a virtual user input device. One such system includes a video camera, memory and a processor. The video camera is configured to produce a series of video frames. The memory is configured to store program code, and the processor is programmed by the program code to examine at least one of the video frames to identify at least one location of a fingertip within the video frame; and map the location to a one of a plurality of predetermined user inputs, based on the location of the fingertip relative to a virtual user input device.

DETAILED DESCRIPTION

The embodiments disclosed herein provide systems and methods for camera-based fingertip tracking. In one such embodiment, a video camera captures motion of one or more fingertips in a series of video frames. Software analyzes the video frames to identify a fingertip location and/or movement in the video frames. Based on location of the fingertip relative to a virtual user input device, the software maps the fingertip to a virtual input such as a key or mouse location, and provides this input to an application.

FIG. 1 depicts one embodiment of a system for camera-based fingertip tracking. System 100 includes a computer 110 in communication with a digital video camera 120. Video camera 120 captures digital images in its field of view, and fingertip tracking logic 250 (see FIG. 2) analyzes series of captured images to track the location and motion of fingertips. Logic 250 then maps the fingertip location and/or motion to data that is usable by an application running on computer 110. In some embodiments, logic 250 maps fingertip location and/or motion to a key, a combination of keys, a mouse click, or a mouse movement.

In the example of FIG. 1, video camera 120 is located near computer 110, with at least one fingertip 130 positioned in the field of view of video camera 120. However, other locations for video camera 120 are possible, as long as video camera 120 is in communication with computer 110. In this example embodiment, video camera 120 is separate from computer 110, but in other embodiments video camera 120 is integrated with computer 110.

Also shown in FIG. 1 is a display 140 (optional). In some embodiments, computer 110 presents a keyboard representation on display 140. The user points to locations on the keyboard representation, fingertip tracking logic 250 tracks the location and motion of fingertips relative to this keyboard representation, then maps the location and/or motion to keyboard locations. In other embodiments, computer 110 projects a keyboard representation onto a surface. The user’s fingers are not required to touch the display or the surface in these embodiments. Instead, the user’s interaction is with the representation of the keyboard rather than a physical object, since fingertip tracking logic 250 detects fingertip location and tracks fingertip motion rather than touch or contact.

FIG. 2 is a hardware block diagram of one embodiment of computer 110. Computer 110 contains a number of components that are familiar to a person of ordinary skill in the art. FIG. 2 omits a number of conventional components, known to those skilled in the art, that are not necessary to explain the operation of the computer.

Components of computer 110 include a processor 210, memory 220, and storage 230 (e.g., hard disk, flash RAM, flash ROM, EEPROM, etc.). These components are coupled via a bus 240. Bus 240 may include multiple types of buses such as a dedicated memory bus, a combined memory-IO bus, and/or one or more peripheral buses (e.g., universal serial bus, IEEE 1394, etc.). The computer is also in communication with video camera 120. In the example of FIG. 2, video camera 120 is connected via bus 240. However, in other embodiments, computer 110 includes a network interface (not shown), and video camera 120 communicates with computer 110 over a network (e.g., Ethernet, WiFi, Internet, etc.).

In some embodiments, fingertip tracking logic 250 is implemented as executable instructions stored in memory.
motion by subtracting the pixel grayscale from the second target criteria in these later frames, the target pixel then

more detail. At least a portion of pixel grid

the previous frame.

from video camera 120, where each frame is processed as a pixel grid 310, shown in FIG. 3. At least one portion of the pixels within pixel grid 310 are identified by logic 250 as a fingertip 130 (see FIG. 1). In some embodiments, logic 250 identifies multiple fingertips 130, each corresponding to a different portion of pixel grid 310.

FIG. 5 is a flowchart of the operation in one embodiment of fingertip tracking logic 250. A positioning stage (block 410) compares each pixel in a current frame with the same pixel in the previous frame. If the grayscale (or color, if color is used) of the pixel has changed significantly, the current pixel is then compared with other neighboring pixels. A pixel with changed grayscale that is also brighter than its surroundings is considered to be a "target" pixel since fingertip 130 tends to be brighter than its surrounding area. Determining that a particular target pixel may correspond to fingertip 130 thus locates the potential fingertip's location in space. Various embodiments may track different numbers of fingertips.

Once fingertips have been identified via confirmed pixels, this embodiment of fingertip tracking logic 250 maps fingertips 130 from locations within pixel grid 310 to user input (block 450), and directs the user input to applications running on the computer 110 (block 460). The techniques described herein allow tracking of all fingertips independently and in free space, which permits greater flexibility and provides more information from gestures.

FIG. 5 illustrates the operation of mapping block 450 in more detail. At least a portion of pixel grid 310 corresponds to a virtual keyboard 510. Each key 520 of virtual keyboard 510 corresponds to a set of pixel locations within grid 310. In the example of FIG. 5, the number "1" key (310) is located at grid location [3,3]. This embodiment of fingertip tracking logic 250 therefore maps fingertip 130 to the "1" key on keyboard 510 when that fingertip 130 is located at location grid location [3,3]. In some embodiments, a key mapping is not determined until a fingertip 130 has stayed in the same location for a specific period of time.

Other fingertip positions are similarly mapped to other locations on virtual keyboard 510. The virtual keyboard concept can be extended to handle key sequences on a physical keyboard as virtual keys. For example, the Ctrl-X key sequence, which represents a "Cut" command in some user interfaces, can be represented as a single "Cut" key on virtual keyboard 510.

Another example of mapping fingertip motion to user input treats fingertip motion as movement of a pointing device (e.g., mouse, trackball). Thus, the software converts motion of the index fingertip in free space to motion of the index finger on the pixel grid, which is mapped to moving a pointer on a virtual surface. One embodiment uses a special location on the grid to represent mouse actions such as a button click or double click, so that moving the index finger to this location acts as a click or double click. In some embodiments, the mouse action is not activated until a fingertip 130 has stayed in the same click or double click location for a specific period of time.

FIG. 6 depicts one example architecture which can be used to implement fingertip tracking logic 250. Device drivers for particular input devices provide low-level events to an operating system 610. This example includes a mouse driver 620 and a keyboard driver 630, which provide mouse events 640 and key events 650 to operating system 610. Operating system 610 then provides user interface events 660, 670 to various applications 680 executing on computer 110. Typically, operating system 610 translates from low-level events to corresponding user interface events, though this translation may not be present in all embodiments. For example, the key sequence "Control-A" may be received by operating system 610 as two keystrokes, but translated into a single user interface event for the application.

A camera driver 690 interfaces to the video camera 120 of FIG. 1, and fingertip tracking logic 250 retrieves frames from camera driver 690. Logic 250 determines fingertip locations and/or motion, then translates location and motion information (relative to the frame or grid) into user input. (This operation was described earlier in connection with FIGS. 3-5.) This user input, in the form of mouse-specific or keyboard-specific data, is provided to mouse driver 620 and keyboard driver 630 (respectively). Thus, in this example architecture fingertip tracking logic 250 operates as a mouse and a keyboard from the perspective of the mouse and keyboard drivers. Some embodiments of fingertip tracking logic 250 may also use an application-specific mapping 695 (e.g., specific to a word processor, spreadsheet, etc.).

FIG. 7 depicts another example architecture which can be used to implement fingertip tracking logic 250. This architecture is similar to that of FIG. 6, but fingertip tracking logic 250 itself operates as a mouse driver (by providing a mouse driver interface 710) and as a keyboard driver (by providing a keyboard driver interface 720). That is, fingertip tracking logic 250 provides the same interfaces to operating system 610 that mouse driver 620 and keyboard driver 630 do.

Embodiments of fingertip tracking logic 250 can control robots from a remote location, which may be used in many different fields and applications. One example is medical diagnostic and surgical procedures. Another example is military applications, for example a targeting system. The techniques described herein can be extended through the use of multiple cameras to implement a three-dimensional tracking system, in which fingertip movement is tracked in three coordinate planes.

In some embodiments system 100 is located in an operating room and used by a person such as a surgeon. Such embodiments include a display which displays, in the field of
The following: an electrical connection (electronic) having
connection with an instruction execution system, apparatus, or
to a software application.

The system of claim 7, wherein the virtual user input
device is a mouse.

9. The system of claim 7, wherein the virtual user input
device is associated with an application-specific mapping,

10. The system of claim 7, further comprising a second
video camera and a third video camera, each of the cameras
producing a corresponding series of video frames, wherein
the processor is further programmed to: examine a series of
video frames from each of the cameras to identify at least one
location of a fingertip within a three-dimensional space
defined by the cameras; and map the location to a one of a
plurality of predetermined user inputs, based on the location
of the fingertip relative to a three dimensional virtual user
input device.

11. A system comprising: a display; a video camera con-
figured to produce a series of video frames; memory config-
ured to store program code thereon; and a processor that is
programmed to: provide the user input to a software applica-
tion, wherein the processor is further programmed to: provide
the user input to a software application.

12. A method of determining user input from a series of
video frames, the method comprising: identifying at least one
location of a fingertip in at least one of the video frames; and
mapping the location to a user input based on the location
of the fingertip relative to a virtual user input device, wherein
said identifying at least one location of a fingertip comprises
identifying a target pixel in a first one of the video frames, the
target pixel representing a potential fingertip location; track-
ing motion of the target pixel in video frames occurring after
the first one; determining whether the target pixel is a con-
firmed pixel, the confirmed pixel representing a probable
fingertip location; and tracking motion of the confirmed pixel
to produce a plurality of fingertip locations.

2. The method of claim 1, wherein the virtual user input
device is a keyboard.

3. The method of claim 1, wherein the virtual user input
device is a mouse.

4. The method of claim 1, further comprising: providing
the user input to a software application.

5. The method of claim 1, wherein the identifying further
comprises: identifying the at least one location of a fingertip
based on presence of a particular color.

6. The method of claim 1, wherein the identifying further
comprises: identifying the at least one location of a fingertip
based on presence of fluorescence.

7. A system comprising: a video camera configured to
produce a series of video frames; memory configured to store
program code thereon; and a processor that is programmed by
the program code to: examine at least one of the video frames
to identify at least one location of a fingertip within the video
frame; and map the location to one of a plurality of predetermined
user inputs, based on the location of the fingertip relative to a virtual user input device, wherein the processor is
further programmed to: identify a target pixel in a first one of
the video frames, the target pixel representing a potential
fingertip location; track motion of the target pixel in video frames occurring after the first one; determine whether the
target pixel is a confirmed pixel, the confirmed pixel represent-
ing a probable fingertip location; and track motion of the
confirmed pixel to produce a plurality of fingertip locations.

8. The system of claim 7, wherein the processor is further
programmed to: provide the user input to a software applica-
tion.

9. The system of claim 7, wherein the virtual user input
device is associated with an application-specific mapping,
and the processor is further programmed to map the location
in accordance with the application-specific mapping.

10. The system of claim 7, further comprising a second
video camera and a third video camera, each of the cameras
producing a corresponding series of video frames, wherein
the processor is further programmed to: examine a series of
video frames from each of the cameras to identify at least one
location of a fingertip within a three-dimensional space
defined by the cameras; and map the location to a one of a
plurality of predetermined user inputs, based on the location
of the fingertip relative to a three dimensional virtual user
input device.

11. A system comprising: a display; a video camera con-
figured to produce a series of video frames; memory config-
ured to store program code thereon; and a processor that is
programmed by the program code to: examine at least one of
the video frames to identify at least one location of a fingertip
within the video frame; and map the location to one of a
plurality of predetermined user inputs, based on the location
of the fingertip relative to a virtual user input device; use the
mapped predetermined user input to control an application
producing a window on the display; the processor being further programmed to: identifying a target pixel in a first one of the video frames, the target pixel representing a potential fingertip location; tracking motion of the target pixel in video frames occurring after the first one; determine whether the target pixel is a confirmed pixel, the confirmed pixel representing a probable fingertip location; and track motion of the confirmed pixel to produce a plurality of fingertip locations.

12. The system of claim 11, wherein the application is an image viewer, the window corresponds to the image, and the mapped predetermined input controls manipulation of the image.

13. The system of claim 11, wherein the application is a browser.

* * * * *