



- (51) **International Patent Classification:**
G06F 9/44 (2006.01)
- (21) **International Application Number:**
PCT/CN2015/090852
- (22) **International Filing Date:**
26 September 2015 (26.09.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Ms: RNB-4-150, Santa Clara, California 95054 (US).
- (72) **Inventors; and**
- (71) **Applicants (for BZ only):** SHEN, Haihao [CN/CN]; No. 880, Zixing Road, Minhang District, Shanghai 200241 (CN). WANG, Yun [CN/CN]; No. 880, Zixing Road, Minhang District, Shanghai 200241 (CN). CHENG, Wen-hua [CN/CN]; No. 880, Zixing Road, Minhang District, Shanghai 200241 (CN). GONG, Jiong [CN/CN]; No. 880, Zixing Road, Minhang District, Shanghai 200241 (CN).
- (74) **Agent:** NTD PATENT AND TRADEMARK AGENCY LIMITED; 10th Floor, Block A, Investment Plaza, 27 Jin-rongdajie, Xicheng District, Beijing 100033 (CN).

- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) **Title:** TECHNOLOGIES FOR AUTOMATED APPLICATION EXPLORATORY TESTING

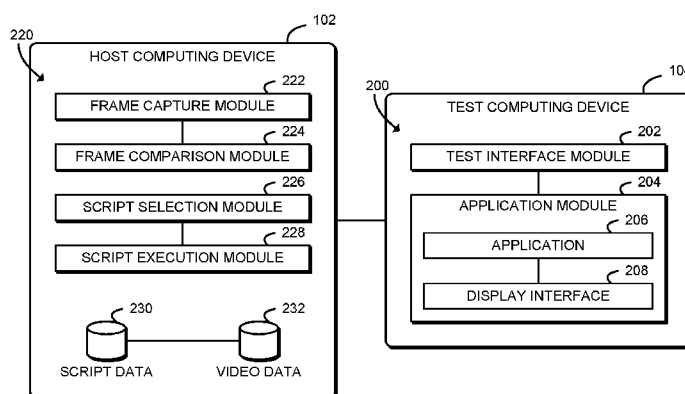


FIG. 2

(57) **Abstract:** Technologies for automated application exploratory testing include a host computing device and a test computing device. The host computing device captures a frame of a display interface of a test application executed by the test computing device. The host computing device determines whether the captured frame matches video data associated with predefined test scripts. The host computing device may detect still features of the video data and compare the captured frame to the still features. The host computing device may schedule a test script for execution that is associated with a matching video. The host computing device may continue executing a currently executing test script if it is associated with a matching video. If no video matches the captured frame, the host computing device performs exploratory execution of the application and may dismiss a transient user interface. Other embodiments are described and claimed.

TECHNOLOGIES FOR AUTOMATED APPLICATION EXPLORATORY TESTING

BACKGROUND

[0001] Currently, a wide number and variety of applications (“apps”) are available for many different computing platforms. However, as the number of applications and the number of computing devices that may execute those applications increase, application validation and testing becomes an increasingly difficult problem. Application validation and testing may require a tester to design a test case that mimics real-world human interaction with an application. In some cases, the tester may be required to validate the same application on multiple platforms, for example validating the application against multiple computing devices with different form factors (e.g., screen size, aspect ratio, etc.). In some cases, the tester may be required to validate the application in dynamic scenarios involving network usage, application upgrades, or other dynamic interactions. Creation and/or maintenance of such tests may require large amounts of manual effort.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The concepts described herein are illustrated by way of example and not by way of limitation in the accompanying figures. For simplicity and clarity of illustration, elements illustrated in the figures are not necessarily drawn to scale. Where considered appropriate, reference labels have been repeated among the figures to indicate corresponding or analogous elements.

[0003] FIG. 1 is a simplified block diagram of at least one embodiment of a system for automated application exploratory testing;

[0004] FIG. 2 is a simplified block diagram of at least one embodiment of various environments that may be established by the system of FIG. 1;

[0005] FIG. 3 is a schematic diagram illustrating at least one embodiment of a method for automated application exploratory testing that may be executed by a computing device of FIGS. 1 and 2; and

[0006] FIG. 4 is a simplified flow diagram of at least one embodiment of a method for automated application exploratory testing that may be executed by a computing device of FIGS. 1 and 2.

DETAILED DESCRIPTION OF THE DRAWINGS

[0007] While the concepts of the present disclosure are susceptible to various modifications and alternative forms, specific embodiments thereof have been shown by way of example in the drawings and will be described herein in detail. It should be understood, however, that there is no intent to limit the concepts of the present disclosure to the particular forms disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives consistent with the present disclosure and the appended claims.

[0008] References in the specification to “one embodiment,” “an embodiment,” “an illustrative embodiment,” etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may or may not necessarily include that particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to effect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described. Additionally, it should be appreciated that items included in a list in the form of “at least one of A, B, and C” can mean (A); (B); (C); (A and B); (A and C); (B and C); or (A, B, and C). Similarly, items listed in the form of “at least one of A, B, or C” can mean (A); (B); (C); (A and B); (A and C); (B and C); or (A, B, and C).

[0009] The disclosed embodiments may be implemented, in some cases, in hardware, firmware, software, or any combination thereof. The disclosed embodiments may also be implemented as instructions carried by or stored on one or more transitory or non-transitory machine-readable (e.g., computer-readable) storage media, which may be read and executed by one or more processors. A machine-readable storage medium may be embodied as any storage device, mechanism, or other physical structure for storing or transmitting information in a form readable by a machine (e.g., a volatile or non-volatile memory, a media disc, or other media device).

[0010] In the drawings, some structural or method features may be shown in specific arrangements and/or orderings. However, it should be appreciated that such specific arrangements and/or orderings may not be required. Rather, in some embodiments, such features may be arranged in a different manner and/or order than shown in the illustrative figures. Additionally, the inclusion of a structural or method feature in a particular figure is not meant to

imply that such feature is required in all embodiments and, in some embodiments, may not be included or may be combined with other features.

[0011] Referring now to FIG. 1, in an illustrative embodiment, a system 100 for automated application exploratory testing includes a host computing device 102 and one or more test computing devices 104, which may be in communication over a network 106. In use, as described in more detail below, the host computing device 102 connects to a test computing device 104 and starts an application test session. The system 100 executes an application in one of three automatic testing modes: (1) using a predefined scripted critical path through the application, (2) selecting among multiple scripted paths, and (3) performing exploratory execution for dynamic scenarios. The system 100 selects an appropriate automatic testing mode and appropriate scripted testing paths based on image analysis of the display interface of the application under test. Thus, the system 100 may reduce the manual effort required to create and/or maintain test scripts, and may enable application testing at large scales. For example, the system 100 may reduce the manual effort required for application testing by 3-5 times (i.e., allowing a tester to test 3-5 times more applications with the same effort).

[0012] The host computing device 102 may be embodied as any type of computation or computer device capable of performing the functions described herein, including, without limitation, a computer, a desktop computer, a workstation, a laptop computer, a notebook computer, a tablet computer, a mobile computing device, a wearable computing device, a network appliance, a web appliance, a distributed computing system, a processor-based system, and/or a consumer electronic device. As shown in FIG. 1, the host computing device 102 illustratively includes a processor 120, an input/output subsystem 122, a memory 124, a data storage device 126, and communication circuitry 128. Of course, the host computing device 102 may include other or additional components, such as those commonly found in a desktop computer (e.g., various input/output devices), in other embodiments. Additionally, in some embodiments, one or more of the illustrative components may be incorporated in, or otherwise form a portion of, another component. For example, the memory 124, or portions thereof, may be incorporated in the processor 120 in some embodiments.

[0013] The processor 120 may be embodied as any type of processor capable of performing the functions described herein. The processor 120 may be embodied as a single or multi-core processor(s), digital signal processor, microcontroller, or other processor or

processing/controlling circuit. Similarly, the memory 124 may be embodied as any type of volatile or non-volatile memory or data storage capable of performing the functions described herein. In operation, the memory 124 may store various data and software used during operation of the host computing device 102 such as operating systems, applications, programs, libraries, and drivers. The memory 124 is communicatively coupled to the processor 120 via the I/O subsystem 122, which may be embodied as circuitry and/or components to facilitate input/output operations with the processor 120, the memory 124, and other components of the host computing device 102. For example, the I/O subsystem 122 may be embodied as, or otherwise include, memory controller hubs, input/output control hubs, firmware devices, communication links (i.e., point-to-point links, bus links, wires, cables, light guides, printed circuit board traces, etc.) and/or other components and subsystems to facilitate the input/output operations. In some embodiments, the I/O subsystem 122 may form a portion of a system-on-a-chip (SoC) and be incorporated, along with the processors 120, the memory 124, and other components of the host computing device 102, on a single integrated circuit chip.

[0014] The data storage device 126 may be embodied as any type of device or devices configured for short-term or long-term storage of data such as, for example, memory devices and circuits, memory cards, hard disk drives, solid-state drives, or other data storage devices. The communication circuitry 128 of the host computing device 102 may be embodied as any communication circuit, device, or collection thereof, capable of enabling communications between the host computing device 102, the test computing devices 104, and/or other remote devices either directly or over the network 106. The communication circuitry 128 may be configured to use any one or more communication technology (e.g., wired or wireless communications) and associated protocols (e.g., direct serial communication, USB communication, Ethernet, Bluetooth®, Wi-Fi®, WiMAX, etc.) to effect such communication.

[0015] Additionally, the host computing device 102 may also include a display 130 and a camera 132. The display 130 may be embodied as any type of display capable of displaying digital information such as a liquid crystal display (LCD), a light emitting diode (LED), a plasma display, a cathode ray tube (CRT), or other type of display device. As described below, the display 130 may be used to display a graphical user interface or other information to the user of the host computing device 102. Additionally, in some embodiments, the host computing device 102 may include a touch screen coupled to the display 130. The touch screen may be used to

record user input that is similar to user input of the test computing device 104, as described further below.

[0016] The camera 132 may be embodied as a digital camera or other digital imaging device integrated with the host computing device 102 or otherwise communicatively coupled thereto. The camera 132 includes an electronic image sensor, such as an active-pixel sensor (APS), e.g., a complementary metal-oxide-semiconductor (CMOS) sensor, or a charge-coupled device (CCD). The camera 132 may be used to capture images of the user interface presented by one or more of the test computing devices 104 including, in some embodiments, capturing still images or video images.

[0017] Each of the test computing devices 104 is configured to execute an application under test and, in some embodiments, provide data on the interface of the application to the host computing device 102 and/or respond to commands initiated by the host computing device 102. Each test computing device 104 may be embodied as any type of computation or computer device capable of performing the functions described herein, including, without limitation, a mobile computing device, a smartphone, a tablet computer, a wearable computing device, a computer, a laptop computer, a desktop computer, multiprocessor system, a server, a rack-mounted server, a blade server, a network appliance, a web appliance, a distributed computing system, a processor-based system, and/or a consumer electronic device. Each test computing device 104 may include components and devices commonly found in a smartphone or similar computing device, such as a processor 140, an I/O subsystem 142, a memory 144, a data storage device 146, communication circuitry 148, a display 150, and/or other peripheral devices. Those individual components of the test computing device 104 may be similar to the corresponding components of the host computing device 102, the description of which is applicable to the corresponding components of the test computing device 104 and is not repeated herein so as not to obscure the present disclosure.

[0018] Additionally, in some embodiments, each test computing device 104 may include a touch screen 152. The touch screen 152 may be embodied as any type of touch screen capable of generating input data in response to being touched by the user of the test computing device 104. The touch screen 152 may be embodied as, for example, a resistive touch screen, a capacitive touch screen, or a camera-based touch screen.

[0019] As discussed in more detail below, the host computing device 102 and the test computing devices 104 may be configured to transmit and receive data with each other and/or other devices of the system 100 over the network 106. The network 106 may be embodied as any number of various wired and/or wireless networks. For example, the network 106 may be embodied as, or otherwise include, a wired or wireless local area network (LAN), a wired or wireless wide area network (WAN), a cellular network, and/or a publicly-accessible, global network such as the Internet. As such, the network 106 may include any number of additional devices, such as additional computers, routers, and switches, to facilitate communications among the devices of the system 100. Additionally or alternatively, the host computing device 102 may communicate directly with one or more test computing devices 104, for example over a direct serial connection, direct USB connection, direct wireless connection, or other direct connection.

[0020] Although illustrated as including separate test computing devices 104, it should be understood that in some embodiments, the functions of one or more of the test computing device 104 may be performed by the host computing device 102. For example, the host computing device 102 may execute a platform simulator associated with one or more test computing devices 104. As another example, in some embodiments the host computing device 102 may natively execute the application under test.

[0021] Referring now to FIG. 2, in the illustrative embodiment, a test computing device 104 establishes an environment 200 during operation. The illustrative environment 200 includes a test interface module 202 and an application module 204. The various modules of the environment 200 may be embodied as hardware, firmware, software, or a combination thereof. As such, in some embodiments, one or more of the modules of the environment 200 may be embodied as circuitry or collection of electrical devices (e.g., test interface circuitry 202 and/or application circuitry 204). It should be appreciated that, in such embodiments, one or more of the test interface circuitry 202 and/or the application circuitry 204 may form a portion of one or more of the processor 140, the I/O subsystem 142, the memory 144, the data storage 146, the communication circuitry 148, and/or other components of the test computing device 104. Additionally, in some embodiments, one or more of the illustrative modules may form a portion of another module and/or one or more of the illustrative modules may be independent of one another.

[0022] The test interface module 202 is configured to communicate with the host computing device 102 during an application test session. For example, the test interface module 202 may be configured to receive commands from the host computing device 102 to start or stop an application test record session or an application test playback session, and the test interface module 202 may be configured to receive commands from the host computing device 102 corresponding to requested user interface actions. The test interface module 202 may also be configured to transmit information to the host computing device 102, such as user interface event data or display interface data.

[0023] The application module 204 is configured to execute an application 206 during an application test session. In some embodiments, the application module 204 may be configured to control the application 206, for example by issuing synthetic user interface events to the application 206. The application 206 may be embodied as a computer program executed by the test computing device 104 such as a native application, a web application, a bytecode application, or any other executable application. The particular format, underlying operating system or application toolkit, or other characteristics of the application 206 may depend on the particular test computing device 104 that executes the application 206. During execution, the application 206 creates and/or manages a display interface 208, which may be displayed on the display 150 of the test computing device 104. The display interface 208 may be embodied as any graphical user interface, and may include multiple user interface objects, such as buttons, menu items, text labels, images, or other user interface controls. The size, layout, appearance, language, and other characteristics of the display interface 208 may also depend on the particular test computing device 104 that executes the application 206.

[0024] Still referring to FIG. 2, in an illustrative embodiment, the host computing device 102 establishes an environment 220 during operation. The illustrative environment 220 includes a frame capture module 222, a frame comparison module 224, a script selection module 226, and a script execution module 228. The various modules of the environment 220 may be embodied as hardware, firmware, software, or a combination thereof. As such, in some embodiments, one or more of the modules of the environment 220 may be embodied as circuitry or collection of electrical devices (e.g., frame capture circuitry 222, frame comparison circuitry 224, etc.). It should be appreciated that, in such embodiments, one or more of the frame capture circuitry 222, the frame comparison circuitry 224, the script selection circuitry 226, and/or the script execution

circuitry 228 may form a portion of one or more of the processor 120, the I/O subsystem 122, the memory 124, the data storage 126, the communication circuitry 128, and/or other components of the host computing device 102. Additionally, in some embodiments, one or more of the illustrative modules may form a portion of another module and/or one or more of the illustrative modules may be independent of one another.

[0025] The frame capture module 222 is configured to capture a frame of the display interface 208 generated by the application 206 executed by the test computing device 104. The frame capture module 222 may be configured to capture the frame using screen capture software to record framebuffer data or other image data representing the display interface 208. In some embodiments, frame capture module 222 may be configured to capture the frame by recording the contents of the display 150 of the test computing device 104 using the camera 132 of the host computing device 102.

[0026] The frame comparison module 224 is configured to compare the captured frame to one or more reference videos to identify any videos matching the captured frame. Each of the reference videos is associated with a predefined test script. The reference videos may be stored in video data 232, which may include video captured from the display interface 208 of the application 206. For example, the video data 232 may be captured while a tester interacts with the application 206 to record a test script. Each predefined test script may include a sequence of pre-recorded or otherwise predefined user interactions with the application 206. For example, each test script may include a series of user input commands (e.g., taps, clicks, key presses, etc.) and associated timestamp information. The predefined test scripts may be stored in script data 230, which may be as one or more XML files, text files, configuration files, or other data files.

[0027] The script selection module 226 is configured to select one of the reference videos that match the captured frame. The script selection module 226 may be further configured to determine whether a currently executed test script is associated with any video that matches the captured frame and, if so, to select the video associated with the currently executed test script. The script selection module 226 may be further configured to determine whether a single video matches the captured frame and to select the single matching video. Additionally, the script selection module 226 may be configured to determine whether zero videos match the captured frame.

[0028] The script execution module 228 is configured to execute the predefined test script associated with the matching video selected by the script selection module 226. The script execution module 228 is further configured to perform an exploratory execution of the application 206 executed by the test computing device 104 if zero videos match the captured frame. Exploratory execution of the application 206 may include detecting a transient user interface based on the display interface 208 generated by the application 206 and then identifying and activating a close object of the transient user interface.

[0029] Although illustrated as being established by a separate test computing device 104, it should be understood that in some embodiments part or all of the environment 200 may be established by the host computing device 102. For example, in some embodiments, the test interface module 202 and/or the application module 204 may be established by the host computing device 102 natively and/or using a platform simulator associated with one or more test computing devices 104.

[0030] Referring now to FIG. 3, schematic diagram 300 shows a general overview of a method for automatic application exploratory testing that may be performed by the system 100. As shown, a human tester 302 (or multiple testers 302) records several test scripts 304. Each test script 304 represents a user interface session with the application 206, and may include one or more recorded or otherwise predefined user interface events. Each test script 304 is associated with a corresponding video 306, as well as additional information such as timestamp data. During a test playback session, a frame 308 is captured from the display interface 208 of the application 206. A test dispatcher 310 compares the captured frame 308 to the video data 232 associated with the predefined test scripts 304. As described further below, the test dispatcher 308 may select a script 304 associated with a matching video 306 for playback (e.g., script 1 in the illustrative embodiment). If no scripts 304 match the captured frame 308, the test dispatcher 310 may invoke an exploratory execution engine 312 to perform exploratory execution of the application 206, as further described below.

[0031] Referring now to FIG. 4, in use, the host computing device 102 may execute a method 400 for automatic application exploratory testing. The method 400 begins with block 402, in which the host computing device 102 connects to a test computing device 104. As described above, the host computing device 102 may be directly connected to the test computing device 104 using, for example, a direct serial connection, direct USB connection, or direct

wireless connection. In some embodiments, the host computing device 102 may connect to the test computing device 104 using the network 106.

[0032] In block 404, the host computing device 102 starts an application test playback session for an application 206 executed by the test computing device 104. The host computing device 102 may perform any configuration, initialization, or other operations required to cause the test computing device 104 to execute the application 206. For example, the host computing device 102 may side-load or otherwise provide the test computing device 104 with binary code corresponding with the application 206. In some embodiments, the host computing device 102 may issue commands or otherwise cause the test computing device 104 to execute a testing script selected from the script data 230.

[0033] In some embodiments, in block 406, the host computing device 102 may cause the test computing device 104 to launch the application 206. For example, the host computing device 102 may send a message or other command to the test computing device 104 to launch the application 206. In some embodiments, the test computing device 104 may launch the application 206 in a special testing mode or including special testing code (e.g., debugging mode, instrumented execution, etc.). Additionally or alternatively, in some embodiments a user may manually launch the application 206 on the test computing device 104.

[0034] In block 408, the host computing device 102 executes an application test script to test the application 206. For example, the host computing device 102 may execute an initial script selected to start testing of the application 206. As described further below, the host computing device 102 may select different scripts for execution based on the contents of the display interface 208 of the application 206. The host computing device 102 may use any appropriate technique to execute the selected script. For example, the host computing device 102 may perform one or more user interface actions specified by commands of the selected test script. The host computing device 102 may perform any appropriate user interface action described in the test script command(s). For example, the host computing device 102 may perform a touch event, perform a mouse click, perform a key press, wait for a predetermined time, or perform another user interface action. In some embodiments, the host computing device 102 may cause the test computing device 104 to generate a synthetic user interface event, for example by transmitting a command to the test computing device 104. Additionally or alternatively, the host computing device 102 may perform the user interface action by operating the test computing

device 104 using a robotic actuator. For example, the host computing device 102 may touch the user interface object on the touch screen 152 of the test computing device 104 using a robotic finger.

[0035] In block 410, the host computing device 102 captures a frame of the display interface 208 generated by the application 206 of the test computing device 104. The host computing device 102 may use any technique to capture the frame of the display interface 208. For example, the host computing device 102 and/or the test computing device 104 may capture the frame using screen capture software to record framebuffer data or other image data representing the display interface 208. In some embodiments, the host computing device 102 may capture the frame by recording the contents of the display 150 of the test computing device 104 using the camera 132 of the host computing device 102.

[0036] In block 412, the host computing device 102 matches the captured frame against the video data 232 associated with the pre-recorded script data 230. The host computing device 102 may compare the captured frame to the video data 232 to identify videos that include images (e.g., user interface elements, text, graphics, and other images) that match the captured frame. In some embodiments, the host computing device 102 may process the video data 232 to identify static images, that is, static features of the video data 232 that do not change significantly from frame to frame. In many circumstances, the static images may include user interface controls such as buttons and menus that are typically selected by the user. The host computing device 102 may compare the captured frame to the static images extracted from the video data 232 using any appropriate image matching algorithm(s) to identify matching videos. As described above, each matching video of the video data 232 is associated with a predefined test script of the script data 230.

[0037] In block 414, the host computing device 102 determines whether any test scripts are associated with videos that match the captured frame. If not, the method 400 branches ahead to block 424, described below. If any test scripts match the captured frame, the method 400 advances to block 416, in which the host computing device 102 determines whether more than one test script matches the captured frame. If not (i.e., if only one test script matches the current frame), the method 400 loops back to block 408 to execute the only test script that matches the current frame. If more than one test script matches the captured frame, the method 400 advances to block 418.

[0038] In block 418, the host computing device 102 determines whether the currently executing test script matches the captured frame. As described above, the host computing device 102 may cause the test computing device 104 to execute one or more user interface operations based on the currently executing test script. In block 420, the host computing device 102 checks whether the currently executing test script is associated with video data 232 matching the captured frame. If so, the method 400 loops back to block 408 to continue executing the currently executing test script. If the currently executed test script does not match the captured frame, the method 400 advances to block 422.

[0039] In block 422, the host computing device 102 selects a test script from the script data 230 that is associated with video data 232 matching the current frame. The host computing device 102 may randomly select a test script from the test scripts matching the current frame. After selecting the test script, the host computing device 102 schedules the selected test script for execution. For example, the host computing device 102 may transmit the test script to the test computing device 104 for execution or otherwise prepare the selected test script for execution. After scheduling the selected test script for execution, the method 400 loops back to block 408 to execute the selected test script.

[0040] Referring back to block 414, if no scripts match the captured frame, the method 400 branches ahead to block 424, in which the host computing device 102 performs exploratory execution of the test application 206. The host computing device 102 performs exploratory execution to handle dynamic user interface scenarios, such as advertisements, upgrade prompts, networking errors, and other dynamic scenarios not included in the pre-recorded test scripts. In block 426, the host computing device 102 compares the captured frame to predecessor and/or successor frames to detect a transient user interface. If the captured frame does not match the predecessor and/or successor frames, then the captured frame may represent a transient interface such as a popup dialog box, popup advertisement, or other transient feature.

[0041] In block 428, the host computing device 102 detects and activates a “close” object in the transient interface. The host computing device 102 may use one or more image matching algorithms to match buttons, menu items, or other user interface controls typically used to dismiss transient interfaces. For example, the host computing device 102 may detect buttons or other user interface controls including an “X,” a cross, or other symbol typically used to identify close buttons. As another example, the host computing device 102 may detect buttons or other

user interface controls including a descriptive text label such as “close,” “cancel,” or other descriptive text.

[0042] After identifying the close object, the host computing device 102 may perform any appropriate user interface action to activate the close object. For example, the host computing device 102 may perform a touch event, perform a mouse click, perform a key press, or perform another user interface action. In some embodiments, the host computing device 102 may cause the test computing device 104 to generate a synthetic user interface event, for example by transmitting a command to the test computing device 104. Additionally or alternatively, the host computing device 102 may perform the user interface action by operating the test computing device 104 using a robotic actuator. For example, the host computing device 102 may touch the user interface object on the touch screen 152 of the test computing device 104 using a robotic finger. After performing exploratory execution of the test application 206, the method 400 loops back to block 408 to continue executing the application test script.

[0043] It should be appreciated that, in some embodiments, the method 400 may be embodied as various instructions stored on a computer-readable media, which may be executed by the processor 120, a peripheral device, and/or other components of the host computing device 102 to cause the host computing device 102 to perform the corresponding method 400. The computer-readable media may be embodied as any type of media capable of being read by the host computing device 102 including, but not limited to, the memory 124, the data storage 126, a local memory of the processor 120, other memory or data storage devices of the host computing device 102, portable media readable by a peripheral device of the host computing device 102, and/or other media.

[0044] Similarly, it should be appreciated that, in some embodiments, the method 400 may be embodied as various instructions stored on a computer-readable media, which may be executed by the processor 140, a peripheral device, and/or other components of the test computing device 104 to cause the test computing device 104 to perform the corresponding method 400. The computer-readable media may be embodied as any type of media capable of being read by the test computing device 104 including, but not limited to, the memory 144, the data storage 146, a local memory of the processor 140, other memory or data storage devices of the test computing device 104, portable media readable by a peripheral device of the test computing device 104, and/or other media.

EXAMPLES

[0045] Illustrative examples of the technologies disclosed herein are provided below. An embodiment of the technologies may include any one or more, and any combination of, the examples described below.

[0046] Example 1 includes a computing device for automated application testing, the computing device comprising frame capture circuitry to capture a frame of a display interface generated by an application executed by a test computing device; frame comparison circuitry to compare the frame to one or more reference videos to identify zero or more matching videos of the one or more reference videos, wherein each reference video is associated with a predefined test script, and wherein each matching video matches the frame; script selection circuitry to select a predefined test script associated with a selected matching video of the matching videos; and script execution circuitry to execute the predefined test script associated with the selected matching video.

[0047] Example 2 includes the subject matter of Example 1, and wherein to compare the frame to the one or more reference videos comprises to identify a static feature of a reference video.

[0048] Example 3 includes the subject matter of any of Examples 1 and 2, and wherein to comparing the frame to the one or more reference videos comprises to compare the frame to the static feature of the reference video.

[0049] Example 4 includes the subject matter of any of Examples 1-3, and wherein the script selection circuitry is further to determine whether a currently executed test script is associated with a matching video of the matching videos; to select the predefined test script comprises to select the currently executed test script in response to a determination that the currently executed test script is associated with a matching video of the matching videos; and to execute the predefined test script associated with the selected matching video comprises to continue to execute the currently executed test script.

[0050] Example 5 includes the subject matter of any of Examples 1-4, and wherein to select the predefined test script comprises to select the predefined test script associated with the selected matching video from the matching videos in response to a determination that the currently executed test script is not associated with a matching video of the matching videos.

[0051] Example 6 includes the subject matter of any of Examples 1-5, and wherein to select the predefined test script associated with the selected matching video from the matching videos comprises to select the selected matching video randomly from the matching videos.

[0052] Example 7 includes the subject matter of any of Examples 1-6, and wherein the script selection circuitry is further to determine whether the zero or more matching videos include a single matching video; wherein to select the predefined test script comprises to select the predefined test script associated with the single matching video in response to a determination that the zero or more matching videos include the single matching video.

[0053] Example 8 includes the subject matter of any of Examples 1-7, and wherein the script selection circuitry is further to determine whether the zero or more matching videos include zero matching videos; and the script execution circuitry is further to perform an exploratory execution of the application executed by the test computing device in response to a determination that the zero or more matching videos include zero matching videos.

[0054] Example 9 includes the subject matter of any of Examples 1-8, and wherein to perform the exploratory execution of the application executed by the test computing device comprises to detect a transient user interface based on the display interface generated by the application.

[0055] Example 10 includes the subject matter of any of Examples 1-9, and wherein to perform the exploratory execution of the application executed by the test computing device further comprises to identify a close object of the transient user interface; and activate the close object of the transient user interface.

[0056] Example 11 includes the subject matter of any of Examples 1-10, and wherein the computing device comprises the test computing device.

[0057] Example 12 includes a method for automated application testing, the method comprising capturing, by a computing device, a frame of a display interface generated by an application executed by a test computing device; comparing, by the computing device, the frame to one or more reference videos to identify zero or more matching videos of the one or more reference videos, wherein each reference video is associated with a predefined test script, and wherein each matching video matches the frame; selecting, by the computing device, a predefined test script associated with a selected matching video of the matching videos; and

executing, by the computing device, the predefined test script associated with the selected matching video.

[0058] Example 13 includes the subject matter of Example 12, and wherein comparing the frame to the one or more reference videos comprises identifying a static feature of a reference video.

[0059] Example 14 includes the subject matter of any of Examples 12 and 13, and wherein comparing the frame to the one or more reference videos comprises comparing the frame to the static feature of the reference video.

[0060] Example 15 includes the subject matter of any of Examples 12-14, and further including determining whether a currently executed test script is associated with a matching video of the matching videos; wherein selecting the predefined test script comprises selecting the currently executed test script in response to determining that the currently executed test script is associated with a matching video of the matching videos; and wherein executing the predefined test script associated with the selected matching video comprises continuing to execute the currently executed test script.

[0061] Example 16 includes the subject matter of any of Examples 12-15, and wherein selecting the predefined test script comprises selecting the predefined test script associated with the selected matching video from the matching videos in response to determining that the currently executed test script is not associated with a matching video of the matching videos.

[0062] Example 17 includes the subject matter of any of Examples 12-16, and wherein selecting the predefined test script associated with the selected matching video from the matching videos comprises randomly selecting the selected matching video from the matching videos.

[0063] Example 18 includes the subject matter of any of Examples 12-17, and further including determining, by the computing device, whether the zero or more matching videos include a single matching video; wherein selecting the predefined test script comprises selecting the predefined test script associated with the single matching video in response to determining that the zero or more matching videos include the single matching video.

[0064] Example 19 includes the subject matter of any of Examples 12-18, and further including determining, by the computing device, whether the zero or more matching videos include zero matching videos; and performing, by the computing device, an exploratory

execution of the application executed by the test computing device in response to determining that the zero or more matching videos include zero matching videos.

[0065] Example 20 includes the subject matter of any of Examples 12-19, and wherein performing the exploratory execution of the application executed by the test computing device comprises detecting, by the computing device, a transient user interface based on the display interface generated by the application.

[0066] Example 21 includes the subject matter of any of Examples 12-20, and wherein performing the exploratory execution of the application executed by the test computing device further comprises identifying a close object of the transient user interface; and activating the close object of the transient user interface.

[0067] Example 22 includes the subject matter of any of Examples 12-21, and wherein the computing device comprises the test computing device.

[0068] Example 23 includes a computing device comprising a processor; and a memory having stored therein a plurality of instructions that when executed by the processor cause the computing device to perform the method of any of Examples 12-22.

[0069] Example 24 includes one or more machine readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a computing device performing the method of any of Examples 12-22.

[0070] Example 25 includes a computing device comprising means for performing the method of any of Examples 12-22.

[0071] Example 26 includes a computing device for automated application testing, the computing device comprising means for capturing a frame of a display interface generated by an application executed by a test computing device; means for comparing the frame to one or more reference videos to identify zero or more matching videos of the one or more reference videos, wherein each reference video is associated with a predefined test script, and wherein each matching video matches the frame; selecting a predefined test script associated with a selected matching video of the matching videos; and means for executing the predefined test script associated with the selected matching video.

[0072] Example 27 includes the subject matter of Example 26, and wherein the means for comparing the frame to the one or more reference videos comprises means for identifying a static feature of a reference video.

[0073] Example 28 includes the subject matter of any of Examples 26 and 27, and wherein the means for comparing the frame to the one or more reference videos comprises means for comparing the frame to the static feature of the reference video.

[0074] Example 29 includes the subject matter of any of Examples 26-28, and further including means for determining whether a currently executed test script is associated with a matching video of the matching videos; wherein the means for selecting the predefined test script comprises means for selecting the currently executed test script in response to determining that the currently executed test script is associated with a matching video of the matching videos; and wherein the means for executing the predefined test script associated with the selected matching video comprises means for continuing to execute the currently executed test script.

[0075] Example 30 includes the subject matter of any of Examples 26-29, and wherein the means for selecting the predefined test script comprises means for selecting the predefined test script associated with the selected matching video from the matching videos in response to determining that the currently executed test script is not associated with a matching video of the matching videos.

[0076] Example 31 includes the subject matter of any of Examples 26-30, and wherein the means for selecting the predefined test script associated with the selected matching video from the matching videos comprises means for randomly selecting the selected matching video from the matching videos.

[0077] Example 32 includes the subject matter of any of Examples 26-31, and further including means for determining whether the zero or more matching videos include a single matching video; wherein the means for selecting the predefined test script comprises means for selecting the predefined test script associated with the single matching video in response to determining that the zero or more matching videos include the single matching video.

[0078] Example 33 includes the subject matter of any of Examples 26-32, and further including means for determining whether the zero or more matching videos include zero matching videos; and means for performing an exploratory execution of the application executed by the test computing device in response to determining that the zero or more matching videos include zero matching videos.

[0079] Example 34 includes the subject matter of any of Examples 26-33, and wherein the means for performing the exploratory execution of the application executed by the test

computing device comprises means for detecting a transient user interface based on the display interface generated by the application.

[0080] Example 35 includes the subject matter of any of Examples 26-34, and wherein the means for performing the exploratory execution of the application executed by the test computing device further comprises means for identifying a close object of the transient user interface; and means for activating the close object of the transient user interface.

[0081] Example 36 includes the subject matter of any of Examples 26-35, and wherein the computing device comprises the test computing device.

CLAIMS:

1. A computing device for automated application testing, the computing device comprising:

frame capture circuitry to capture a frame of a display interface generated by an application executed by a test computing device;

frame comparison circuitry to compare the frame to one or more reference videos to identify zero or more matching videos of the one or more reference videos, wherein each reference video is associated with a predefined test script, and wherein each matching video matches the frame;

script selection circuitry to select a predefined test script associated with a selected matching video of the matching videos; and

script execution circuitry to execute the predefined test script associated with the selected matching video.

2. The computing device of claim 1, wherein to compare the frame to the one or more reference videos comprises to identify a static feature of a reference video.

3. The computing device of claim 2, wherein to comparing the frame to the one or more reference videos comprises to compare the frame to the static feature of the reference video.

4. The computing device of claim 1, wherein:
the script selection circuitry is further to determine whether a currently executed test script is associated with a matching video of the matching videos;

to select the predefined test script comprises to select the currently executed test script in response to a determination that the currently executed test script is associated with a matching video of the matching videos; and

to execute the predefined test script associated with the selected matching video comprises to continue to execute the currently executed test script.

5. The computing device of claim 4, wherein to select the predefined test script comprises to select the predefined test script associated with the selected matching video from

the matching videos in response to a determination that the currently executed test script is not associated with a matching video of the matching videos.

6. The computing device of claim 5, wherein to select the predefined test script associated with the selected matching video from the matching videos comprises to select the selected matching video randomly from the matching videos.

7. The computing device of claim 1, wherein:
the script selection circuitry is further to determine whether the zero or more matching videos include a single matching video;

wherein to select the predefined test script comprises to select the predefined test script associated with the single matching video in response to a determination that the zero or more matching videos include the single matching video.

8. The computing device of any of claims 1-7, wherein:
the script selection circuitry is further to determine whether the zero or more matching videos include zero matching videos; and

the script execution circuitry is further to perform an exploratory execution of the application executed by the test computing device in response to a determination that the zero or more matching videos include zero matching videos.

9. The computing device of claim 8, wherein to perform the exploratory execution of the application executed by the test computing device comprises to detect a transient user interface based on the display interface generated by the application.

10. The computing device of claim 9, wherein to perform the exploratory execution of the application executed by the test computing device further comprises to:

identify a close object of the transient user interface; and
activate the close object of the transient user interface.

11. The computing device of any of claims 1-7, wherein the computing device comprises the test computing device.

12. A method for automated application testing, the method comprising:
capturing, by a computing device, a frame of a display interface generated by an application executed by a test computing device;
comparing, by the computing device, the frame to one or more reference videos to identify zero or more matching videos of the one or more reference videos, wherein each reference video is associated with a predefined test script, and wherein each matching video matches the frame;
selecting, by the computing device, a predefined test script associated with a selected matching video of the matching videos; and
executing, by the computing device, the predefined test script associated with the selected matching video.
13. The method of claim 12, wherein comparing the frame to the one or more reference videos comprises identifying a static feature of a reference video.
14. The method of claim 13, wherein comparing the frame to the one or more reference videos comprises comparing the frame to the static feature of the reference video.
15. The method of claim 12, further comprising:
determining whether a currently executed test script is associated with a matching video of the matching videos;
wherein selecting the predefined test script comprises selecting the currently executed test script in response to determining that the currently executed test script is associated with a matching video of the matching videos; and
wherein executing the predefined test script associated with the selected matching video comprises continuing to execute the currently executed test script.
16. The method of claim 15, wherein selecting the predefined test script comprises selecting the predefined test script associated with the selected matching video from the matching videos in response to determining that the currently executed test script is not associated with a matching video of the matching videos.

17. The method of claim 16, wherein selecting the predefined test script associated with the selected matching video from the matching videos comprises randomly selecting the selected matching video from the matching videos.

18. The method of claim 12, further comprising:
determining, by the computing device, whether the zero or more matching videos include a single matching video;
wherein selecting the predefined test script comprises selecting the predefined test script associated with the single matching video in response to determining that the zero or more matching videos include the single matching video.

19. The method of claim 12, further comprising:
determining, by the computing device, whether the zero or more matching videos include zero matching videos; and
performing, by the computing device, an exploratory execution of the application executed by the test computing device in response to determining that the zero or more matching videos include zero matching videos.

20. The method of claim 19, wherein performing the exploratory execution of the application executed by the test computing device comprises detecting, by the computing device, a transient user interface based on the display interface generated by the application.

21. The method of claim 20, wherein performing the exploratory execution of the application executed by the test computing device further comprises:
identifying a close object of the transient user interface; and
activating the close object of the transient user interface.

22. The method of claim 12, wherein the computing device comprises the test computing device.

23. A computing device comprising:
a processor; and

a memory having stored therein a plurality of instructions that when executed by the processor cause the computing device to perform the method of any of claims 12-22.

24. One or more machine readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a computing device performing the method of any of claims 12-22.

25. A computing device comprising means for performing the method of any of claims 12-22.

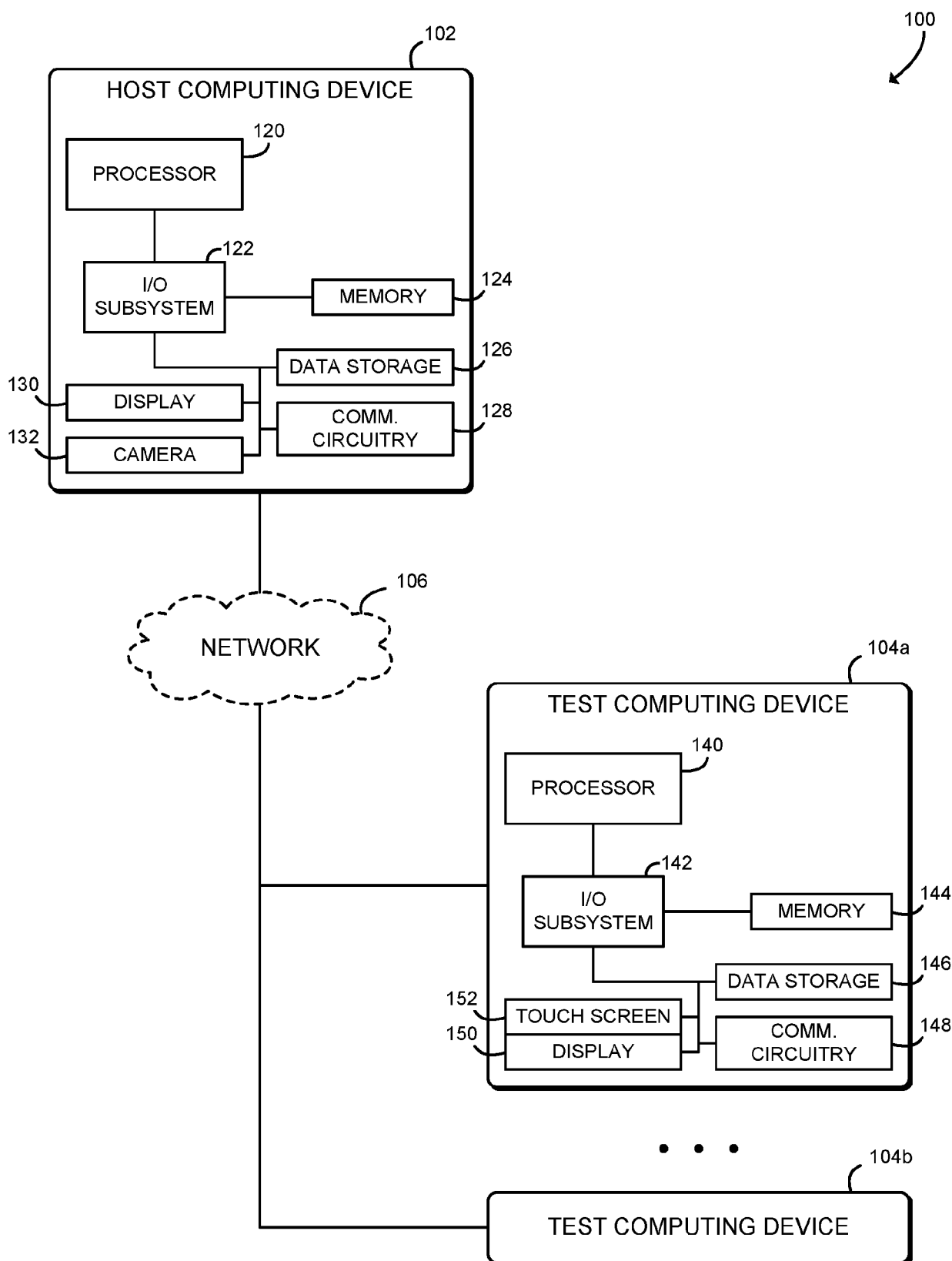


FIG. 1

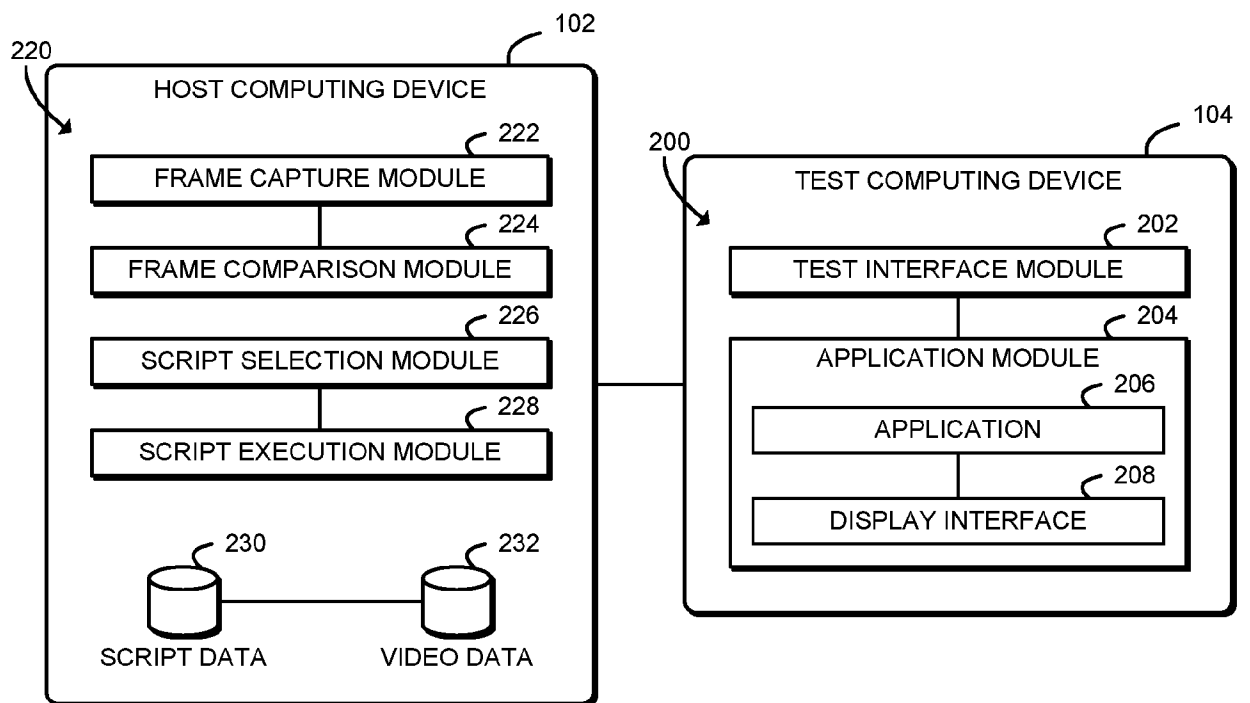


FIG. 2

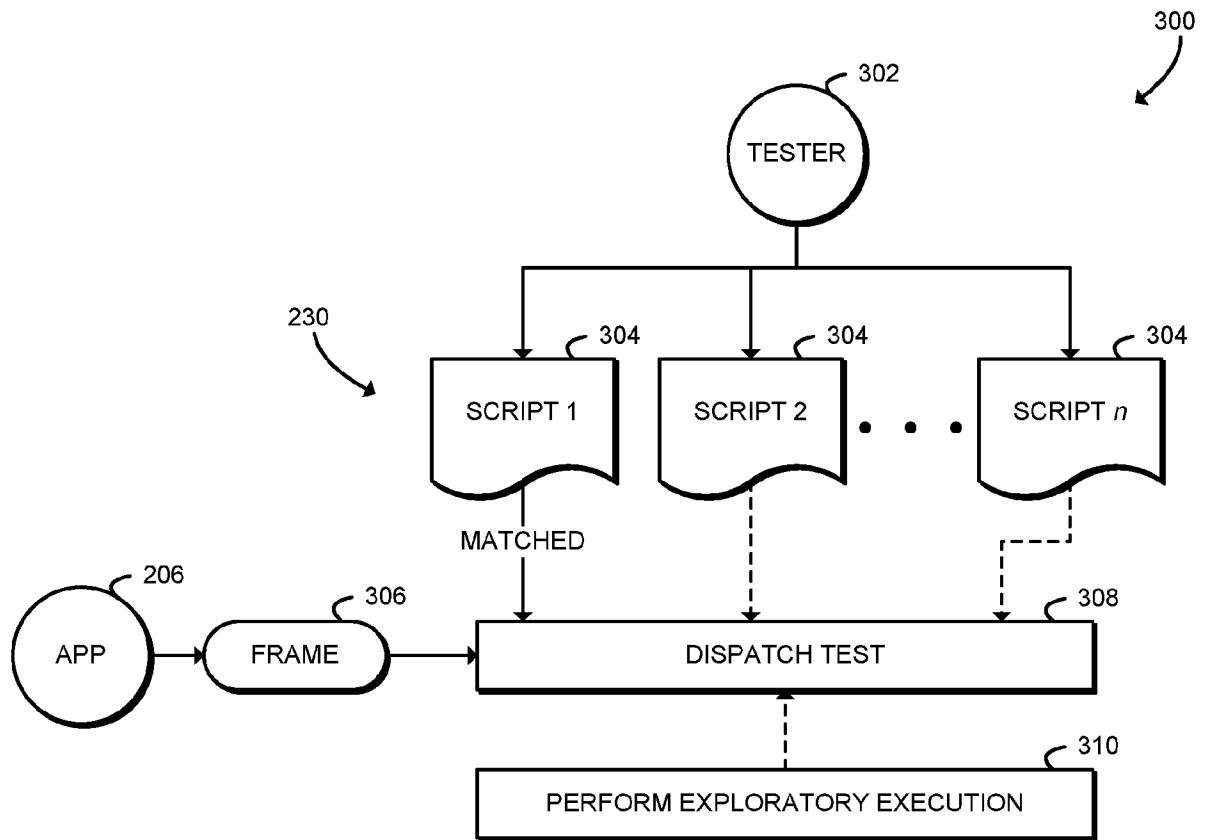


FIG. 3

4/4

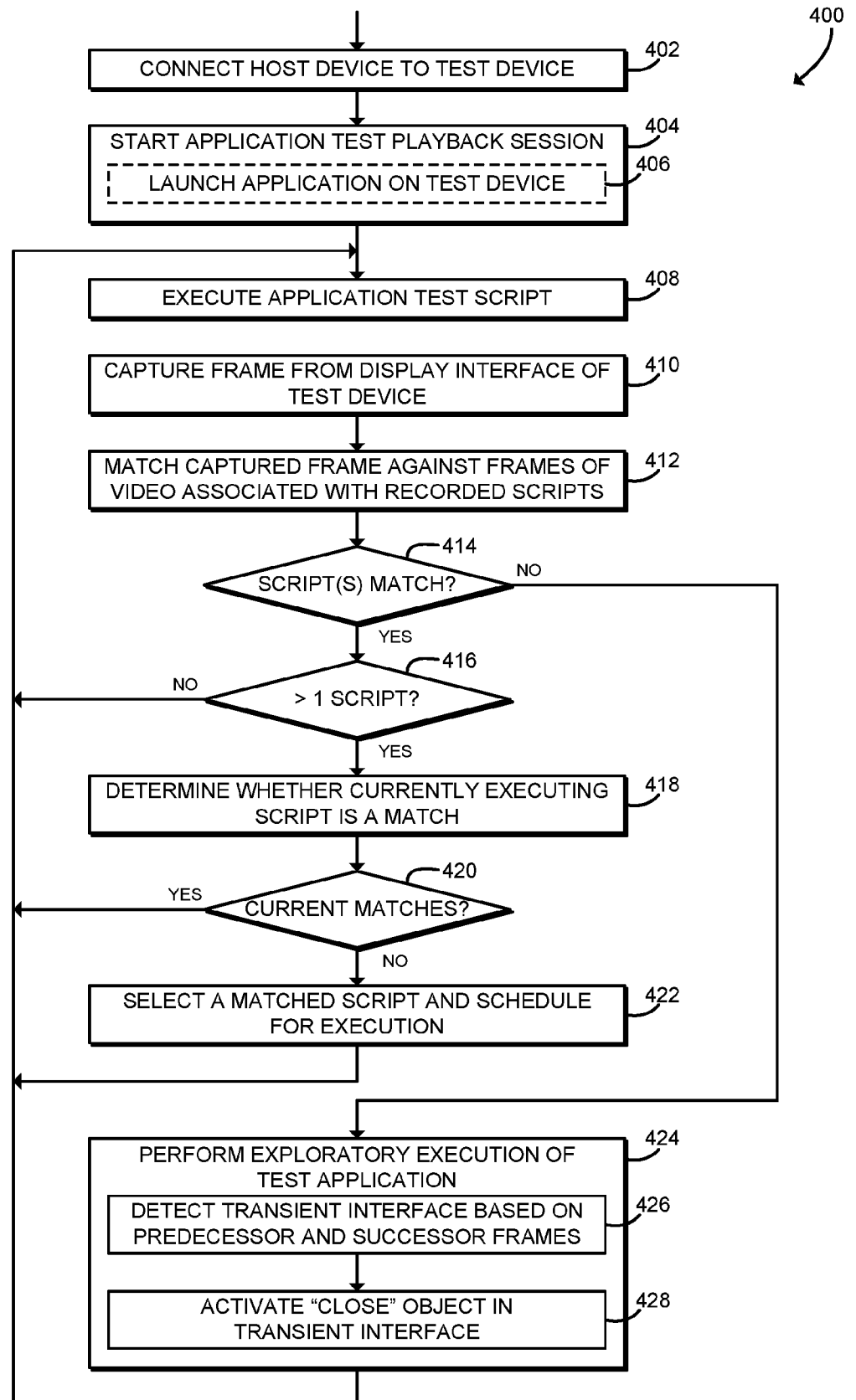


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2015/090852

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/44(2006.01)i

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)

G06F

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

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

CNPAT,CNKI,WPI,EPODOC,IEEE: application, test, script, video, image, match, frame, display, auto

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2013263090 A1 (SONY ONLINE ENTERTAINMENT LLC) 03 October 2013 (2013-10-03) description, paragraphs [0008], [0027]	1-25
A	US 2013290938 A1 (NIR, DOR ET AL.) 31 October 2013 (2013-10-31) the whole document	1-25
A	CN 104572052 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 29 April 2015 (2015-04-29) the whole document	1-25
A	CN 104050076 A (ALIBABA GROUP HOLDING CO., LTD.) 17 September 2014 (2014-09-17) the whole document	1-25



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 application or patent but 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

01 June 2016

Date of mailing of the international search report

23 June 2016

Name and mailing address of the ISA/CN

STATE INTELLECTUAL PROPERTY OFFICE OF THE
P.R.CHINA
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing
100088, China

Facsimile No. (86-10)62019451

Authorized officer

KONG,Xin

Telephone No. (86-10)62413668

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2015/090852

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2013263090	A1	03 October 2013	CN	103365773	A	23 October 2013
US	2013290938	A1	31 October 2013	None			
CN	104572052	A	29 April 2015	None			
CN	104050076	A	25 November 2015	None			