



Office de la Propriété

Intellectuelle  
du Canada

Un organisme  
d'Industrie Canada

Canadian  
Intellectual Property  
Office

An agency of  
Industry Canada

CA 2964471 A1 2016/04/28

(21) **2 964 471**

(12) **DEMANDE DE BREVET CANADIEN**  
**CANADIAN PATENT APPLICATION**

(13) **A1**

(86) Date de dépôt PCT/PCT Filing Date: 2015/10/15  
(87) Date publication PCT/PCT Publication Date: 2016/04/28  
(85) Entrée phase nationale/National Entry: 2017/04/12  
(86) N° demande PCT/PCT Application No.: US 2015/055618  
(87) N° publication PCT/PCT Publication No.: 2016/064642  
(30) Priorité/Priority: 2014/10/22 (US14/521,368)

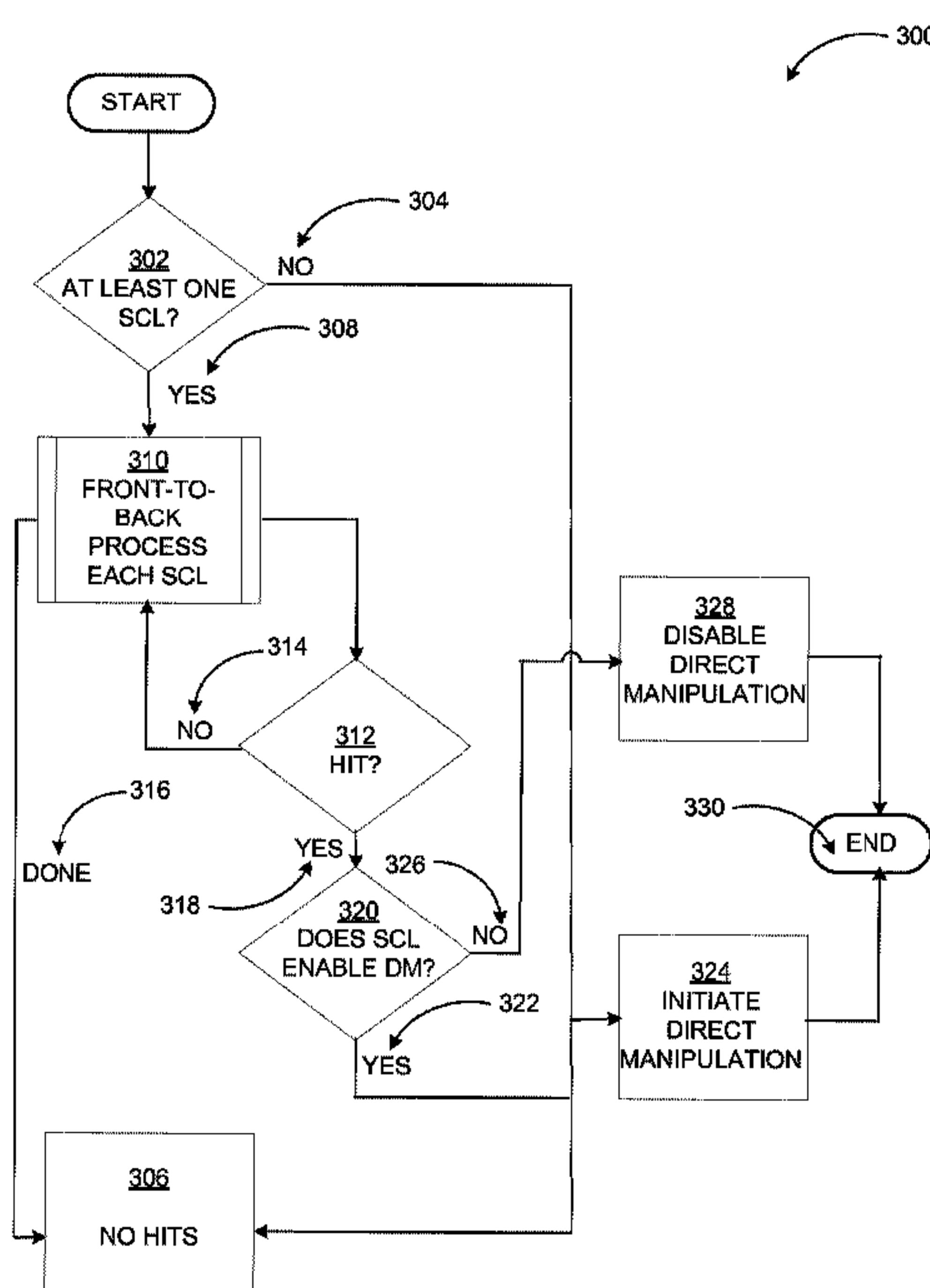
(51) Cl.Int./Int.Cl. *G06F 3/0484* (2013.01),  
*G06F 3/0482* (2013.01)

(71) Demandeur/Applicant:  
MICROSOFT TECHNOLOGY LICENSING, LLC, US

(72) Inventeurs/Inventors:  
HOFSETZ, CHRISTIAN, US;  
EDEN, HEATHER, US;  
KAROLEWICS, STEPHEN, US;  
KRANTZ, JAMES, US;  
DALTON, MICHAEL, US;  
SUN, SIWEN, US;  
YOUNG, KERRY, US

(74) Agent: SMART & BIGGAR

(54) Titre : TEST D'OCCURRENCE PERMETTANT DE DETERMINER L'ACTIVATION DE MANIPULATIONS DIRECTES  
EN REPONSE A DES ACTIONS D'UTILISATEUR  
(54) Title: HIT-TEST TO DETERMINE ENABLEMENT OF DIRECT MANIPULATIONS IN RESPONSE TO USER  
ACTIONS



**FIG. 3**

(57) Abrégé/Abstract:

A hit-test may be performed to determine whether a user action detected via a user interface associated with an application intersects a given object of the application. The application may include one or more special content layers (SCLs). A decision may

**(57) Abrégé(suite)/Abstract(continued):**

be made at each SCL regarding enablement of a direct manipulation of an object through the application in response to the user action. An SCL processing module of the application may identify a hit in response to detecting an intersection of the user action and one or more non-transparent pixels of at least one SCL during a front to back processing of the SCLs, where the pixels correspond to the object. The application may then determine whether a direct manipulation of an object through the application is enabled or not enabled by the SCL, and thus whether the direct manipulation should be automatically initiated or disabled, respectively.

## (12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
28 April 2016 (28.04.2016)

(10) International Publication Number  
**WO 2016/064642 A1**

(51) International Patent Classification:  
**G06F 3/0482** (2013.01)

8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US). **KAROLEWICS, Stephen**; Microsoft Technology Licensing, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US). **KRANTZ, James**; Microsoft Technology Licensing, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US). **DALTON, Michael**; Microsoft Technology Licensing, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US). **SUN, Siwen**; Microsoft Technology Licensing, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US). **YOUNG, Kerry**; Microsoft Technology Licensing, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US).

(21) International Application Number:  
**PCT/US2015/055618**

(22) International Filing Date:  
15 October 2015 (15.10.2015)

(25) Filing Language:  
English

(26) Publication Language:  
English

(30) Priority Data:  
14/521,368 22 October 2014 (22.10.2014) US

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US).

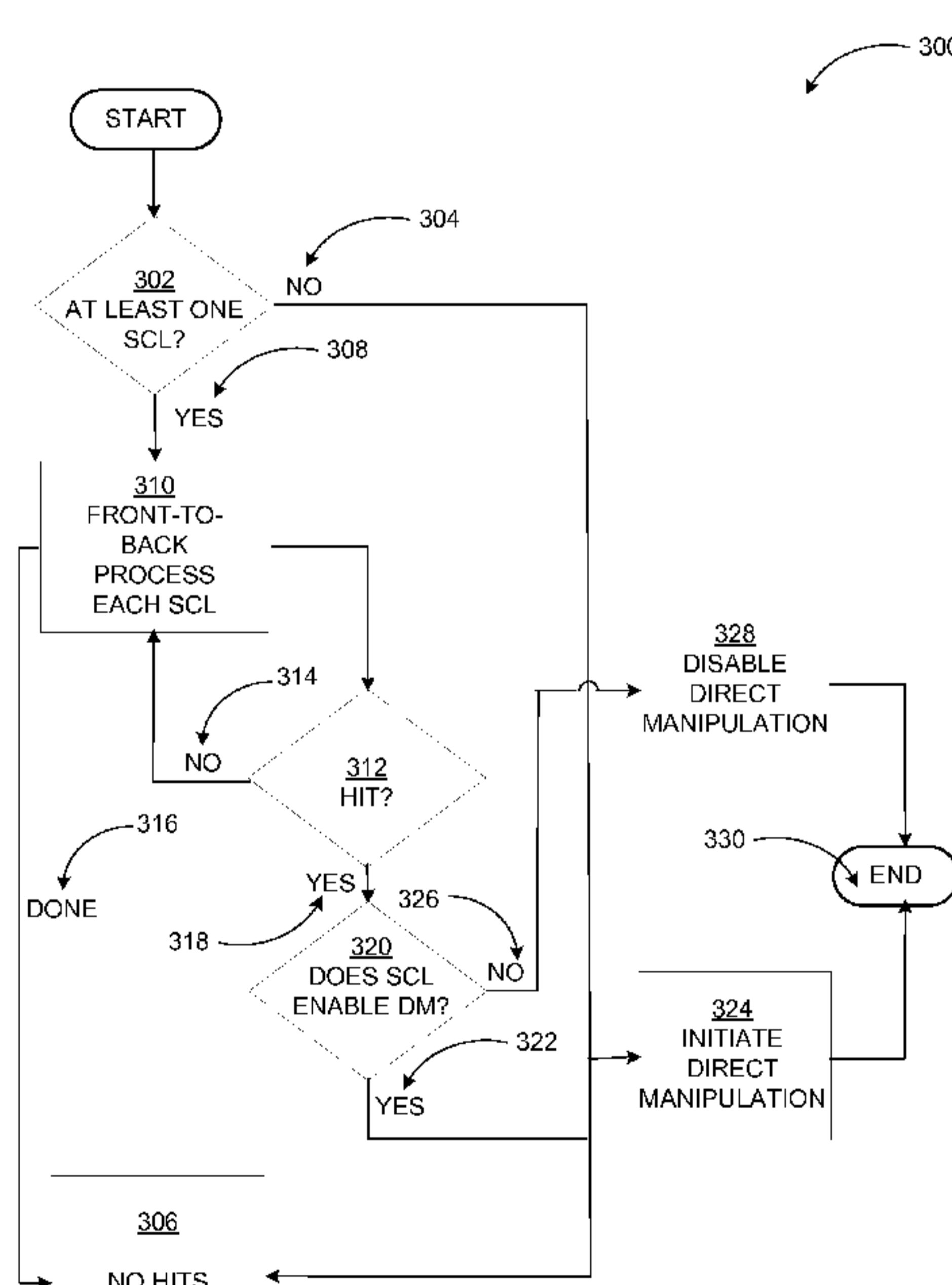
(72) Inventors: **HOFSETZ, Christian**; Microsoft Technology Licensing, LLC, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US). **EDEN, Heather**; Microsoft Technology Licensing, LLC, Attn: Patent Group Docketing (Bldg.

(74) Agents: **MINHAS, Sandip** et al.; Microsoft Corporation, Attn: Patent Group Docketing (Bldg. 8/1000), One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

[Continued on next page]

(54) Title: HIT-TEST TO DETERMINE ENABLEMENT OF DIRECT MANIPULATIONS IN RESPONSE TO USER ACTIONS



(57) Abstract: A hit-test may be performed to determine whether a user action detected via a user interface associated with an application intersects a given object of the application. The application may include one or more special content layers (SCLs). A decision may be made at each SCL regarding enablement of a direct manipulation of an object through the application in response to the user action. An SCL processing module of the application may identify a hit in response to detecting an intersection of the user action and one or more non-transparent pixels of at least one SCL during a front to back processing of the SCLs, where the pixels correspond to the object. The application may then determine whether a direct manipulation of an object through the application is enabled or not enabled by the SCL, and thus whether the direct manipulation should be automatically initiated or disabled, respectively.

**FIG. 3**

## WO 2016/064642 A1



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).

**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

**Published:**

- *with international search report (Art. 21(3))*

## HIT-TEST TO DETERMINE ENABLEMENT OF DIRECT MANIPULATIONS IN RESPONSE TO USER ACTIONS

### BACKGROUND

- 5 [0001] Hit-testing, also known as hit detection, picking, and/or pick correlation, is a process employed to determine whether a user-controlled cursor, such as a mouse cursor or a touch-point on a user interface associated with an application, intersects a given object, such as a shape, line, or curve, of the application displayed on the user interface. Furthermore, hit-testing is employed to respond to user actions, such as selection of a 10 menu item or a target in the application based on its visual location on the user interface.
- [0002] Currently, there are no quick, efficient techniques to determine if content in the application has been hit through either touch input, gesture input, and/or traditional input from the user. Hit-testing may be very expensive and certain types of content associated with the application, such as graphical objects, may take the longest to resolve 15 as the time it takes to calculate hit-testing is bound by a complexity of the content.

### SUMMARY

- [0003] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This summary is not intended to exclusively identify key features or essential features of the claimed subject 20 matter, nor is it intended as an aid in determining the scope of the claimed subject matter.
- [0004] Embodiments are directed to performance of a hit-test to determine enablement of a direct manipulation in response to a user action. A processor of a computing device may be configured to execute an application that includes one or more special content layers (SCLs), and to cause a user interface associated with the application 25 to be presented to a user through a client device upon execution of the application. The application may detect a user action through the user interface, and in response to detecting an intersection of the user action and one or more pixels of at least one SCL of the application, a hit may be identified. Whether a direct manipulation of an object through the application is enabled by the at least one SCL may be determined by the 30 application in response to the hit, and if the direct manipulation of an object through the application is enabled by the at least one SCL, the direct manipulation of an object through the application may be automatically initiated.
- [0005] These and other features and advantages will be apparent from a reading of the following detailed description and a review of the associated drawings. It is to be

understood that both the foregoing general description and the following detailed description are explanatory and do not restrict aspects as claimed.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

**[0006]** FIG. 1 includes an example network environment where a hit-test may be performed;

**[0007]** FIG. 2 illustrates an example user interface associated with an application configured to perform a hit-test;

**[0008]** FIG. 3 illustrates an example process to perform a hit-test to determine enablement of one or more direct manipulations in response to a user action;

**[0009]** FIG. 4 is a block diagram of an example general purpose computing device, which may be used to perform a hit-test to determine enablement of one or more direct manipulations in response to a user action; and

**[0010]** FIG. 5 illustrates a logic flow diagram of a method to perform a hit-test to determine enablement of one or more direct manipulations in response to a user action, according to embodiments.

### **DETAILED DESCRIPTION**

**[0001]** As briefly described above, a hit-test may be performed to determine whether a user action detected via a user interface associated with an application intersects a given object of the application. The application may be a word-processing, spreadsheet, and/or presentation application, for example, and include one or more special content layers (SCLs). As described herein, an SCL is an application layer where an immediate decision regarding enablement of a direct manipulation of an object through the application may be made in response to the user action. An SCL processing module of the application may identify a hit in response to detecting an intersection of the user action and one or more non-transparent pixels of at least one of the SCLs during a front to back processing of the SCLs, where the pixels may correspond to the object of the application. The application may then be configured to determine whether a direct manipulation of an object through the application is enabled by the SCL. If the direct manipulation of an object through the application is enabled by the SCL, the direct manipulation may be automatically initiated. Alternately, if the direct manipulation of an object through the application is not enabled by the SCL, the direct manipulation may be disabled.

**[0002]** In the following detailed description, references are made to the accompanying drawings that form a part hereof, and in which are shown by way of illustrations specific embodiments or examples. These aspects may be combined, other

aspects may be utilized, and structural changes may be made without departing from the spirit or scope of the present disclosure. The following detailed description is therefore not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims and their equivalents.

5 [0003] While some embodiments will be described in the general context of program modules that execute in conjunction with an application program that runs on an operating system on a personal computer, those skilled in the art will recognize that aspects may also be implemented in combination with other program modules.

10 [0004] Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that embodiments may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and comparable computing 15 devices. Embodiments may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

20 [0005] Some embodiments may be implemented as a computer-implemented process (method), a computing system, or as an article of manufacture, such as a computer program product or computer readable media. The computer program product may be a computer storage medium readable by a computer system and encoding a computer program that comprises instructions for causing a computer or computing system to 25 perform example process(es). The computer-readable storage medium is a computer-readable memory device. The computer-readable storage medium can for example be implemented via one or more of a volatile computer memory, a non-volatile memory, a hard drive, a flash drive, a floppy disk, or a compact disk, and comparable hardware media.

30 [0006] Throughout this specification, the term “platform” may be a combination of software and hardware components for hit-test performance. Examples of platforms include, but are not limited to, a hosted service executed over a plurality of servers, an application executed on a single computing device, and comparable systems. The term “server” generally refers to a computing device executing one or more software programs typically in a networked environment. However, a server may also be implemented as a

virtual server (software programs) executed on one or more computing devices viewed as a server on the network. More detail on these technologies and example operations is provided below.

**[0007]** FIG. 1 includes an example network environment where a hit-test may be performed. As demonstrated in diagram 100, one or more users (108, 114, and 120) may access an application 102, such as a word-processing, spreadsheet, and/or presentation application, over a cloud-based network 130. In some examples, the application 102 may include one or more SCLs 104, where an SCL 104 is an application layer where a decision may be made immediately regarding enablement of a direct manipulation of an object through the application 102 in response to a user action intersecting one or more pixels in the SCL 104. For each SCL 104, one or more types of direct manipulation may be defined, and for each type of direct manipulation whether the direct manipulation is enabled by the SCL may further be defined. A number of the SCLs 104 that the application 102 includes may be dependent on a type of the application 102. For example, a presentation application may include a lesser number of SCLs 104 than a word-processing application.

**[0008]** The application 102 may be hosted at a remote server, and may be accessed through a user's client device over the cloud-based network 130. For example, the server may be configured to execute the application 102 and cause a user interface 106 associated with the application 102 to be presented through the user's client device. A local version of the application 102 may also be locally hosted at the user's client device, and data associated with the local application 102 may be retrieved over the cloud-based network 130. Some example client devices may include a desktop computer 122, a laptop computer 110, a smart phone, a car phone, a mobile phone, a tablet 116, and/or a home automation device. For example, a first user 108 may access the application 102 through the laptop computer 110 over the cloud-based network 130, and interact with the user interface 106 using a touch input 112. A second user 114 may access the application 102 through the tablet 116 over the cloud-based network 130, and interact with the user interface 106 using a stylus input 118. A third user 120 may access the application 102 through the desktop computer 122 over the cloud-based network 130, and interact with the user interface 106 using traditional input, such as a mouse 124. Other input methods may include gesture input, and/or keyboard input, for example.

**[0009]** An example application 102 may be configured to detect a user action through the user interface 106. The user action may include the touch input 112, the stylus

input 118, the mouse 124 input, and/or other inputs such as gyroscopic input, eye-tracking, and comparable ones. For example, the touch input 112 of the first user 108, may include a tap action or a swipe action associated with an object, such as a control element, textual element, or graphical element, of the application 102 displayed on the user interface 106.

5 [0010] If the application 102 includes at least one SCL 104, an SCL processing module of the application 102 may be configured to perform front to back processing of the SCL 104 in response to the detected user action. The SCL processing module of the application 102 may be configured to determine whether to perform the front to back processing based on a texture content of an area on the user interface proximal to the user 10 action. For example, if one or more objects of the application (which may include the object associated with the user action, and one or more other objects) are displayed in the area on the user interface proximal to the user action, the texture content corresponding to the objects may indicate to the SCL processing module to perform the front to back processing. Contrastingly, if no objects are displayed in the area on the user interface proximal to the user action, a lack of texture content may indicate to the SCL processing 15 module to not perform the front to back processing, as the user action was likely not intended and a response is not needed. In response to detecting an intersection of the user action and one or more pixels of the SCL 104 during the front to back processing, a hit may be identified. The pixels may be non-transparent pixels that correspond to the object 20 associated with the user action, where at least a portion of the object is located in the area on the user interface 106 proximal to the user action.

[0011] The application 102 may then be configured to determine whether a direct manipulation of an object through the application 102 is enabled by the SCL 104 in response to the hit. If the direct manipulation of an object through the application 102 is 25 enabled by the SCL 104, the direct manipulation of an object through the application 102 may be automatically initiated. If the direct manipulation of an object through the application 102 is not enabled by the SCL 104, the direct manipulation of an object through the application 102 may be disabled. As previously discussed, the SCL 104 is a layer of the application 102 where a decision may be made immediately regarding 30 enablement of a direct manipulation of an object through the application 102. For each SCL 104, one or more types of direct manipulation may be defined, and for each type of direct manipulation whether the direct manipulation is enabled or disabled by the SCL 104 may further be defined. A type of the direct manipulation may be determined based on a behavior of the object relative to the user action, and may include panning, zooming, and

selecting the object, among other examples. For example, if the first user 108 performs a tap action through the touch input 112 on a picture on a document slide of a presentation application, the type of direct manipulation may include enlarging the picture by zooming.

**[0012]** Incorporating the immediate decision capabilities of the SCL 104 with the hit-testing performance may offer a quick, efficient way to determine enablement in response to user action that enhances current hit-testing techniques. For example, the incorporation prevents the current necessity for additional processing steps, and thus prevents the need for further processing software and/or hardware to determine the enablement of direct manipulation, which may reduce an overall operation time and cost.

**[0013]** The example system in FIG. 1 has been described with specific servers, client devices, applications, and interactions. Embodiments are not limited to the system according to this example configuration. A platform for hit-test performance to determine enablement of a direct manipulation may be implemented in configurations employing fewer or additional components, and performing other tasks. Furthermore, platforms for hit-test performance to determine enablement of a direct manipulation may be implemented in a similar manner using the principles described herein.

**[0014]** FIG. 2 illustrates an example user interface associated with an application configured to perform a hit-test. As demonstrated in diagram 200, an application, such as a presentation application, may be accessed by a user through a client device such as a tablet 202. The application may include one or more SCLs, and upon execution of the application, a user interface 204 associated with the application may be presented through the tablet 202.

**[0015]** The application may detect a user action through the user interface 204, where the user action may be performed through a touch input 206 and may include a tap action associated with an object of the application displayed on the user interface 204, such as a graph 208. An SCL processing module of the application may be configured to perform front to back processing of at least one SCL in response to the detection of the user action. The SCL processing module may determine whether to perform the front to back processing based on a texture content of an area on the user interface 204 proximal to the user action. For example, the texture content of the area on the user interface 204 proximal to the tap action may indicate that one or more objects of the application, such as the graph 208 or textual content 210 are proximal, and thus the front to back processing should be performed. In another example, if the user action is a swipe action performed through the touch input in an area on the user interface 204 that does not display any type

of content, the texture content may indicate that no objects are proximal, and thus the front to back processing should not be performed as the user action was likely not intended and thus a response is not needed.

**[0016]** In response to detecting an intersection of the user action and one or more pixels of the SCL, a hit may be identified. The pixels may be non-transparent pixels that correspond to the graph 208 tapped by the user. The application may then be configured to determine whether a direct manipulation of an object through the application is enabled by the SCL in response to the hit, where the direct manipulation may include selecting the graph 208 to perform a copy and paste function, for example. If the direct manipulation of an object through the application is enabled by the SCL, the direct manipulation of an object through the application may be automatically initiated. If the direct manipulation of an object through the application is not enabled by the SCL, the direct manipulation of an object through the application may be disabled.

**[0017]** FIG. 3 illustrates an example process to perform a hit-test to determine enablement of one or more direct manipulations in response to a user action. An application, such as a word-processing, spreadsheet, and/or presentation application, may be accessed by a user through a client device, and upon execution of the application, a user interface associated with the application may be presented through a display of the client device to enable user interaction with the application. In some examples, the application may include one or more SCLs, where a number of the SCLs depends on a type of the application. Each SCL is an application layer where a decision may be made immediately regarding enablement of a direct manipulation of an object through the application in response to a user action.

**[0018]** As demonstrated in a diagram 300, in response to detecting a user action through the user interface associated with the application, an SCL processing module of an application may determine if the application includes at least one SCL 302. If the application does not include at least one SCL 304, no hits may be identified 306. If the application does include at least one SCL 308, an SCL processing module of the application may be configured to perform front to back processing of each SCL 310 in response to the detection of the user action. The SCL processing module may determine whether to perform the front to back processing based on a texture content of an area on the user interface proximal to the user action. The SCL processing module may identify if there are any hits 312 by determining if there is an intersection of the user action and one or more pixels of at least one SCL detected during the front to back processing. If there

are no hits identified 314, the SCL processing module may continue processing other SCLs (i.e., if the application includes more than one SCL) and may complete processing 316 the SCLs once each SCL is processed and no hits are identified 306.

[0019] If there are hits identified 318, the application may determine whether the 5 SCL enables direct manipulation of an object through the application 320. For example, one or more types of direct manipulation may be defined for each SCL, and for each type of direct manipulation whether the direct manipulation is enabled or disabled by the SCL may be further defined. If the SCL enables direct manipulation of an object through the application 322, the direct manipulation may be automatically initiated 324 and the 10 process may end 330. If the SCL does not enable direct manipulation of an object through the application 326, the direct manipulation may be disabled 328 and the process may end 330.

[0020] The examples in FIGs. 1 through 3 have been described using specific 15 network environments, configurations, devices, and processes to perform a hit-test to determine enablement of one or more direct manipulations. Embodiments to perform a hit-test are not limited to the specific network environments, configurations, devices, and processes according to these examples.

[0021] The capability of the one or more SCLs of the application to make an immediate decision regarding enablement or disablement of a direct manipulation of an 20 object in response to a user action may advantageously reduce a load of the processor while simultaneously improving usability in regards to quicker response to user actions.

[0022] FIG. 4 and the associated discussion are intended to provide a brief, general description of a general purpose computing device, which may be used to perform a hit-test to determine enablement of one or more direct manipulations in response to a user 25 action.

[0023] For example, computing device 400 may be used as a server, desktop computer, portable computer, smart phone, special purpose computer, or similar device. In an example basic configuration 402, the computing device 400 may include one or more processors 404 and a system memory 406. A memory bus 408 may be used for 30 communicating between the processor 404 and the system memory 406. The basic configuration 402 is illustrated in FIG. 4 by those components within the inner dashed line.

[0024] Depending on the desired configuration, the processor 404 may be of any type, including but not limited to a microprocessor ( $\mu$ P), a microcontroller ( $\mu$ C), a digital

signal processor (DSP), or any combination thereof. The processor 404 may include one more levels of caching, such as a level cache memory 412, one or more processor cores 414, and registers 416. The example processor cores 414 may (each) include an arithmetic logic unit (ALU), a floating point unit (FPU), a digital signal processing core (DSP Core), or any combination thereof. An example memory controller 418 may also be used with the processor 404, or in some implementations the memory controller 418 may be an internal part of the processor 404.

Depending on the desired configuration, the system memory 406 may be of any type including but not limited to volatile memory (such as RAM), non-volatile memory (such as ROM, flash memory, etc.) or any combination thereof. The system memory 406 may include an operating system 420, an application 422, and program data 424. The application 422 may include one or more SCLs and an SCL processing module 426, which may be an integral part of the application or a separate application on its own. Execution of the application 422 may cause an associated user interface to be presented. In response to detecting a user action through the user interface it may be determined if the application includes at least one SCL. The SCL processing module 426 may perform front to back processing of the at least one SCL of the application to detect an intersection of the user action and the one or more pixels of at least one SCL, which may identify a hit. Whether a direct manipulation of an object through the application is enabled by the at least one SCL may be determined in response to the hit, and if the direct manipulation of an object through the application is enabled by the at least one SCL, the direct manipulation of an object through the application may be automatically initiated. Alternately, if the direct manipulation of an object through the application is not enabled by the at least one SCL, the direct manipulation of an object through the application may be disabled. The program data 424 may include, among other data, process data 428 related to the enablement of the direct manipulation based on SCL and direct manipulation type definitions, as described herein.

**[0025]** The computing device 400 may have additional features or functionality, and additional interfaces to facilitate communications between the basic configuration 402 and any desired devices and interfaces. For example, a bus/interface controller 430 may be used to facilitate communications between the basic configuration 402 and one or more data storage devices 432 via a storage interface bus 434. The data storage devices 432 may be one or more removable storage devices 436, one or more non-removable storage devices 438, or a combination thereof. Examples of the removable storage and the non-

removable storage devices include magnetic disk devices such as flexible disk drives and hard-disk drives (HDDs), optical disk drives such as compact disk (CD) drives or digital versatile disk (DVD) drives, solid state drives (SSD), and tape drives to name a few.

Example computer storage media may include volatile and nonvolatile, removable and

5 non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data.

**[0026]** The system memory 406, the removable storage devices 436 and the non-removable storage devices 438 are examples of computer storage media. Computer 10 storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVDs), solid state drives, or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which may be used to store the desired information and which may be accessed by the computing device 400. Any such 15 computer storage media may be part of the computing device 400.

**[0027]** The computing device 400 may also include an interface bus 440 for facilitating communication from various interface devices (for example, one or more output devices 442, one or more peripheral interfaces 444, and one or more communication devices 446) to the basic configuration 402 via the bus/interface controller 430. Some of the example output devices 442 include a graphics processing unit 448 and an audio processing unit 450, which may be configured to communicate to various 20 external devices such as a display or speakers via one or more A/V ports 452. One or more example peripheral interfaces 444 may include a serial interface controller 454 or a parallel interface controller 456, which may be configured to communicate with external 25 devices such as input devices (for example, keyboard, mouse, pen, voice input device, touch input device, etc.) or other peripheral devices (for example, printer, scanner, etc.) via one or more I/O ports 458. An example communication device 446 includes a network 30 controller 460, which may be arranged to facilitate communications with one or more other computing devices 462 over a network communication link via one or more communication ports 464. The one or more other computing devices 462 may include servers, client devices, and comparable devices.

**[0028]** The network communication link may be one example of a communication media. Communication media may typically be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal,

such as a carrier wave or other transport mechanism, and may include any information delivery media. A “modulated data signal” may be a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media may include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF), microwave, infrared (IR) and other wireless media. The term computer readable media as used herein may include both storage media and communication media.

5 [0029] The computing device 400 may be implemented as a part of a general purpose or specialized server, mainframe, or similar computer that includes any of the above 10 functions. The computing device 400 may also be implemented as a personal computer including both laptop computer and non-laptop computer configurations.

15 [0030] Example embodiments may also include methods to perform a hit-test to determine enablement of one or more direct manipulations. These methods can be implemented in any number of ways, including the structures described herein. One such way may be by machine operations, of devices of the type described in the present disclosure. Another optional way may be for one or more of the individual operations of the methods to be performed in conjunction with one or more human operators performing some of the operations while other operations may be performed by machines. These human operators need not be collocated with each other, but each can be only with a 20 machine that performs a portion of the program. In other embodiments, the human interaction can be automated such as by pre-selected criteria that may be machine automated.

25 [0031] FIG. 5 illustrates a logic flow diagram for process 500 of a method to perform a hit-test to determine enablement of one or more direct manipulations in response to a user action, according to embodiments. Process 500 may be implemented on a server or other system.

30 [0032] Process 500 begins with operation 510, where an application may be configured to detect a user action through a user interface associated with the application. The user action may include a touch input including a tap and swipe action, a gesture input, a pen input, a mouse input, and/or keyboard input, for example. If the application includes at least one SCL, an SCL processing module of the application may be configured to perform front to back processing of each SCL.

[0033] At operation 520, a hit may be identified in response to detecting an intersection of the user action and one or more pixels of the SCL during the front to back

processing. The pixels may be non-transparent pixels that correspond to an object associated with the user action, such as a control element, a textual element, and/or a graphical element, of the application displayed on the user interface. At least a portion, if not all, of the object may be located in an area on the user interface proximal to the user action.

**[0034]** At operation 530, the application may then be configured to determine whether a direct manipulation of an object through the application is enabled by the SCL in response to the hit. The SCL is an application layer where a decision may be made immediately regarding enablement of a direct manipulation of an object through the application. For each SCL, one or more types of direct manipulation may be defined, and for each type of direct manipulation whether the direct manipulation is enabled or disabled by the SCL may further be defined. A type of the direct manipulation may be determined based on a behavior of the object relative to the user action, and may include panning, zooming, and selecting the object, among other examples.

**[0035]** At operation 540, the direct manipulation of an object through the application may be automatically initiated if the direct manipulation of an object through the application is enabled by the SCL. Alternately, if the direct manipulation of an object through the application is not enabled by the SCL, the direct manipulation of an object through the application may be disabled.

**[0036]** The operations included in process 500 are for illustration purposes. Performance of a hit-test to determine enablement of direct manipulations in response to a user action may be implemented by similar processes with fewer or additional steps, as well as in different order of operations using the principles described herein.

**[0037]** According to some embodiments, a method to perform hit-testing is provided. The method may include a means for detecting a user action through a user interface associated with an application, a means for identifying a hit in response to detecting an intersection of the user action and one or more pixels of at least one SCL of the application, a means for determining whether a direct manipulation of an object through the application is enabled by the at least one SCL in response to the hit, and a means for automatically initiating the direct manipulation of an object through the application if the direct manipulation of the object through the application is enabled by the at least one SCL.

**[0038]** According to some examples, methods to perform hit-testing are provided. An example method may include detecting a user action through a user interface

associated with an application, and identifying a hit in response to detecting an intersection of the user action and one or more pixels of at least one SCL of the application. The example method may also include determining whether a direct manipulation of an object through the application is enabled by the at least one SCL in response to the hit, and automatically initiating the direct manipulation of an object through the application if the direct manipulation of the object through the application is enabled by the at least one SCL.

**[0039]** In other examples, the direct manipulation may be disabled if the direct manipulation of the object through the application is not enabled by the at least one SCL.

10 In response to detecting the user action through the user interface associated with the application, it may be determined if the application includes the at least one SCL. Front to back processing of the at least one SCL of the application may be performed at a SCL processing module of the application to detect the intersection of the user action and the one or more pixels of at least one SCL. It may be determined whether to perform the front to back processing based on a texture content of an area on the user interface proximal to the user action.

**[0040]** In further examples, for each SCL one or more types of direct manipulation may be defined. For each type of direct manipulation, whether the direct manipulation is enabled or disabled by the SCL in response to the hit may be defined. The user action may 20 include a touch input, a gesture input, a mouse input, a pen input, an eye-tracking input, a voice command input, a gyroscopic input, and/or a keyboard input.

**[0041]** According to other examples, computing devices to perform hit-tests may be described. An example computing device may include a memory configured to store instructions, and a processor coupled to the memory, the processor executing an 25 application comprising one or more SCLs and causing a user interface associated with the application to be presented. The application may be configured to detect a user action through the user interface associated with the application, and identify a hit in response to detecting an intersection of the user action and one or more pixels of at least one of the SCLs. The application may also be configured to determine whether a direct manipulation 30 of an object through the application is enabled by the at least one SCL in response to the hit, and automatically initiate the direct manipulation of the object through the application if the at least one SCL enables the direct manipulation of the object through the application.

**[0042]** In other embodiments, the one or more pixels of the at least one SCL may be non-transparent. The one or more pixels of the at least one SCL may correspond to the object displayed on the user interface associated with the application. At least a portion of the object may be located in an area on the user interface proximal to the user action. A 5 type of the direct manipulation may be determined based on a behavior of the object relative to the user action, where the type of the direct manipulation includes panning, zooming, and/or selecting.

**[0043]** In further embodiments, a number of the SCLs may be dependent on a type of the application, wherein the application is one or more of a word-processing application, a 10 spreadsheet application, and a presentation application. The user interface associated with the application may be presented to a user through a display of a client device upon execution of the application at the client device.

**[0044]** According to some examples, computer-readable memory devices with 15 instructions stored thereon to perform a hit-test may be described. Example instructions may include detecting a user action through a user interface of an application, and identifying a hit in response to detecting an intersection of the user action and one or more pixels of at least one SCL. The example instructions may also include determining whether a direct manipulation of an object through the application is enabled by the at least one SCL in response to the hit, and automatically initiating the direct manipulation of 20 the object through the application if the direct manipulation of the object through the application is enabled by the at least one SCL, or disabling the direct manipulation if the direct manipulation of the object through the application is not enabled by the at least one SCL.

**[0045]** In other examples, for each SCL one or more types of direct manipulation 25 may be defined. For each type of direct manipulation, if the direct manipulation is enabled or disabled by the SCL in response to the hit may be defined.

**[0046]** The above specification, examples and data provide a complete description of the manufacture and use of the composition of the embodiments. Although the subject matter has been described in language specific to structural features and/or methodological 30 acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims and embodiments.

## CLAIMS

1. A method to perform hit-testing, the method comprising:
  - detecting a user action through a user interface associated with an application;
  - in response to detecting an intersection of the user action and one or more pixels of at least one special content layer (SCL) of the application, identifying a hit;
  - determining whether a direct manipulation of an object through the application is enabled by the at least one SCL in response to the hit; and
  - automatically initiating the direct manipulation of an object through the application if the direct manipulation of the object through the application is enabled by the at least one SCL.
2. The method of claim 1, further comprising:
  - disabling the direct manipulation if the direct manipulation of the object through the application is not enabled by the at least one SCL.
3. The method of claim 1, further comprising:
  - in response to detecting the user action through the user interface associated with the application, determining if the application includes the at least one SCL.
4. The method of claim 1, further comprising:
  - performing front to back processing of the at least one SCL of the application at a SCL processing module of the application to detect the intersection of the user action and the one or more pixels of at least one SCL.
5. The method of claim 4, further comprising:
  - determining whether to perform the front to back processing based on a texture content of an area on the user interface proximal to the user action.
6. The method of claim 1, wherein the user action includes one or more of: a touch input, a gesture input, a mouse input, a pen input, an eye-tracking input, a voice command input, a gyroscopic input, and/or a keyboard input.
7. A computing device to perform a hit-test, the computing device comprising:
  - a memory configured to store instructions; and

a processor coupled to the memory, the processor executing an application comprising one or more special content layers (SCLs) and causing a user interface associated with the application to be presented, wherein the application is configured to:

detect a user action through the user interface associated with the application;

in response to detecting an intersection of the user action and one or more pixels of at least one of the SCLs, identify a hit;

determine whether a direct manipulation of an object through the application is enabled by the at least one SCL in response to the hit; and

automatically initiate the direct manipulation of the object through the application if the at least one SCL enables the direct manipulation of the object through the application.

8. The computing device of claim 7, wherein the one or more pixels of the at least one SCL correspond to the object displayed on the user interface associated with the application.

9. The computing device of claim 8, wherein at least a portion of the object is located in an area on the user interface proximal to the user action.

10. The computing device of claim 8, wherein a type of the direct manipulation is determined based on a behavior of the object relative to the user action, and includes one of panning, zooming, and selecting.

11. The computing device of claim 7, wherein a number of the SCLs is dependent on a type of the application.

12. The computing device of claim 7, wherein the user interface associated with the application is presented to a user through a display of a client device upon execution of the application at the client device.

13. A computer-readable memory device with instructions stored thereon to perform a hit-test, the instructions comprising:

detecting a user action through a user interface of an application;

in response to detecting an intersection of the user action and one or more pixels of at least one special content layer (SCL), identifying a hit;

determining whether a direct manipulation of an object through the application is enabled by the at least one SCL in response to the hit; and

automatically initiating the direct manipulation of the object through the application if the direct manipulation of the object through the application is enabled by the at least one SCL, or

disabling the direct manipulation if the direct manipulation of the object through the application is not enabled by the at least one SCL.

14. The computer-readable memory device of claim 13, wherein the instructions further comprise:

defining, for each SCL, one or more types of direct manipulation.

15. The computer-readable memory device of claim 14, wherein the instructions further comprise:

defining, for each type of direct manipulation, if the direct manipulation is enabled or disabled by the SCL in response to the hit.


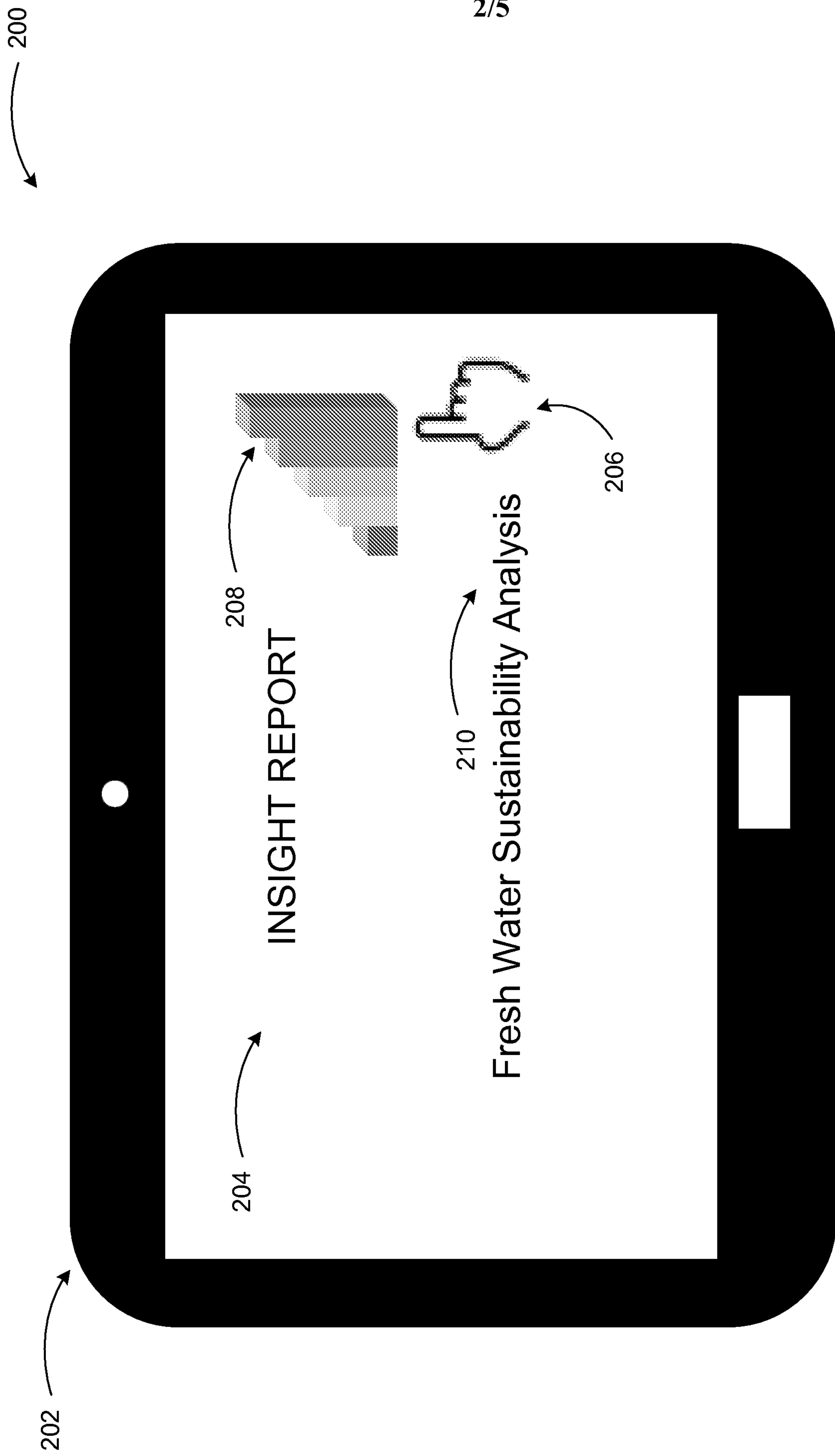


FIG. 1

2/5



**FIG. 2**

3/5

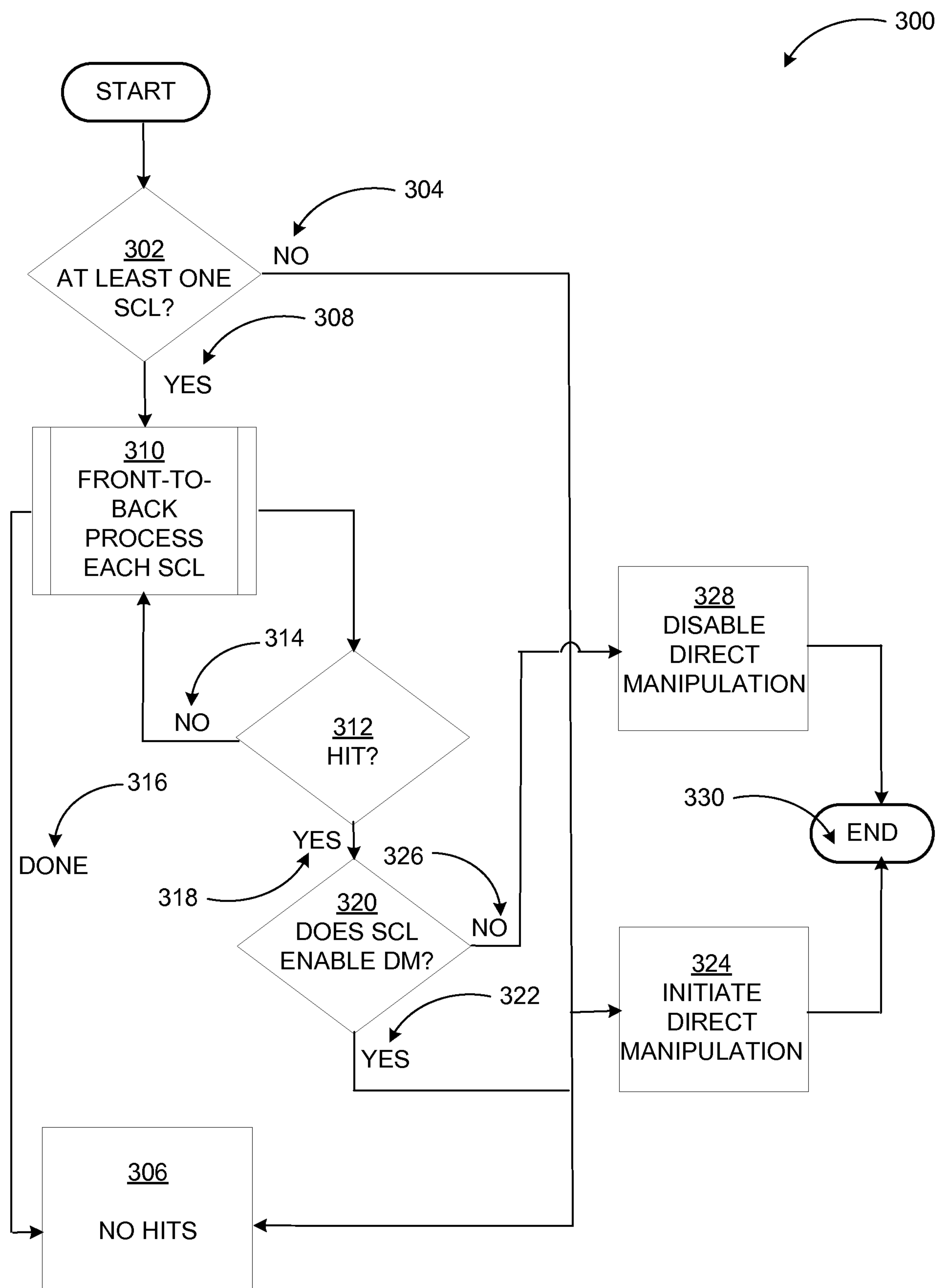
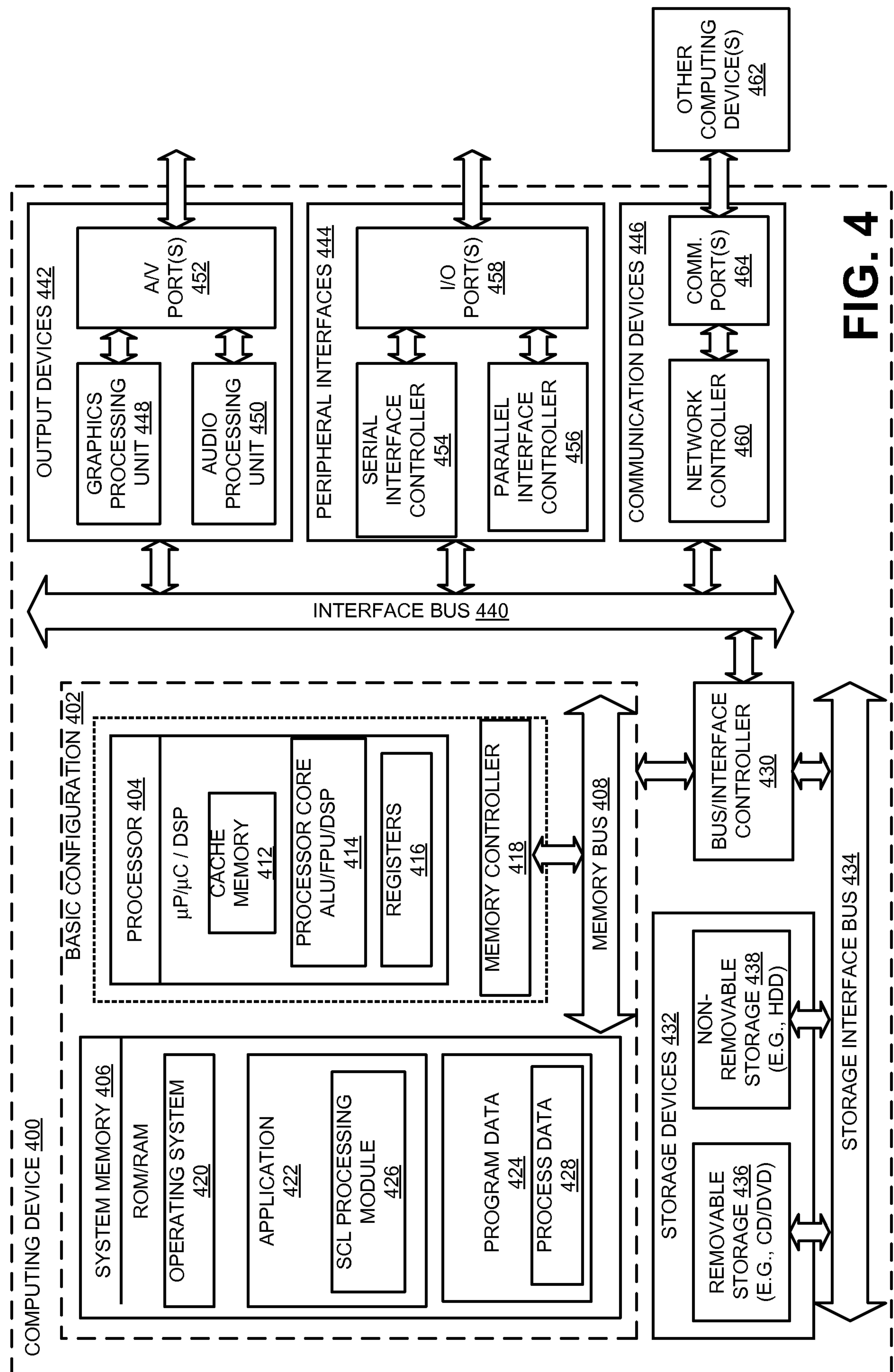
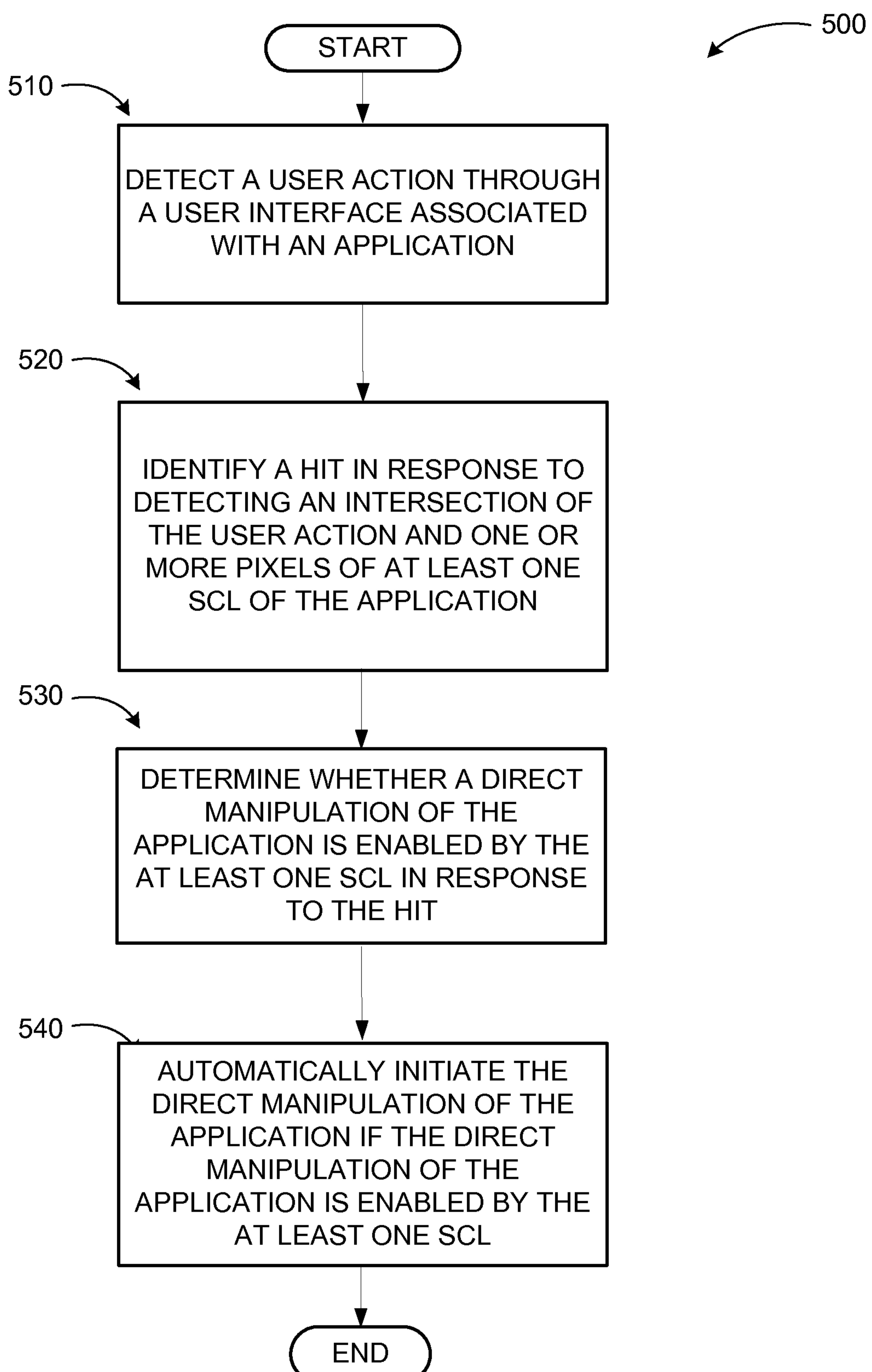


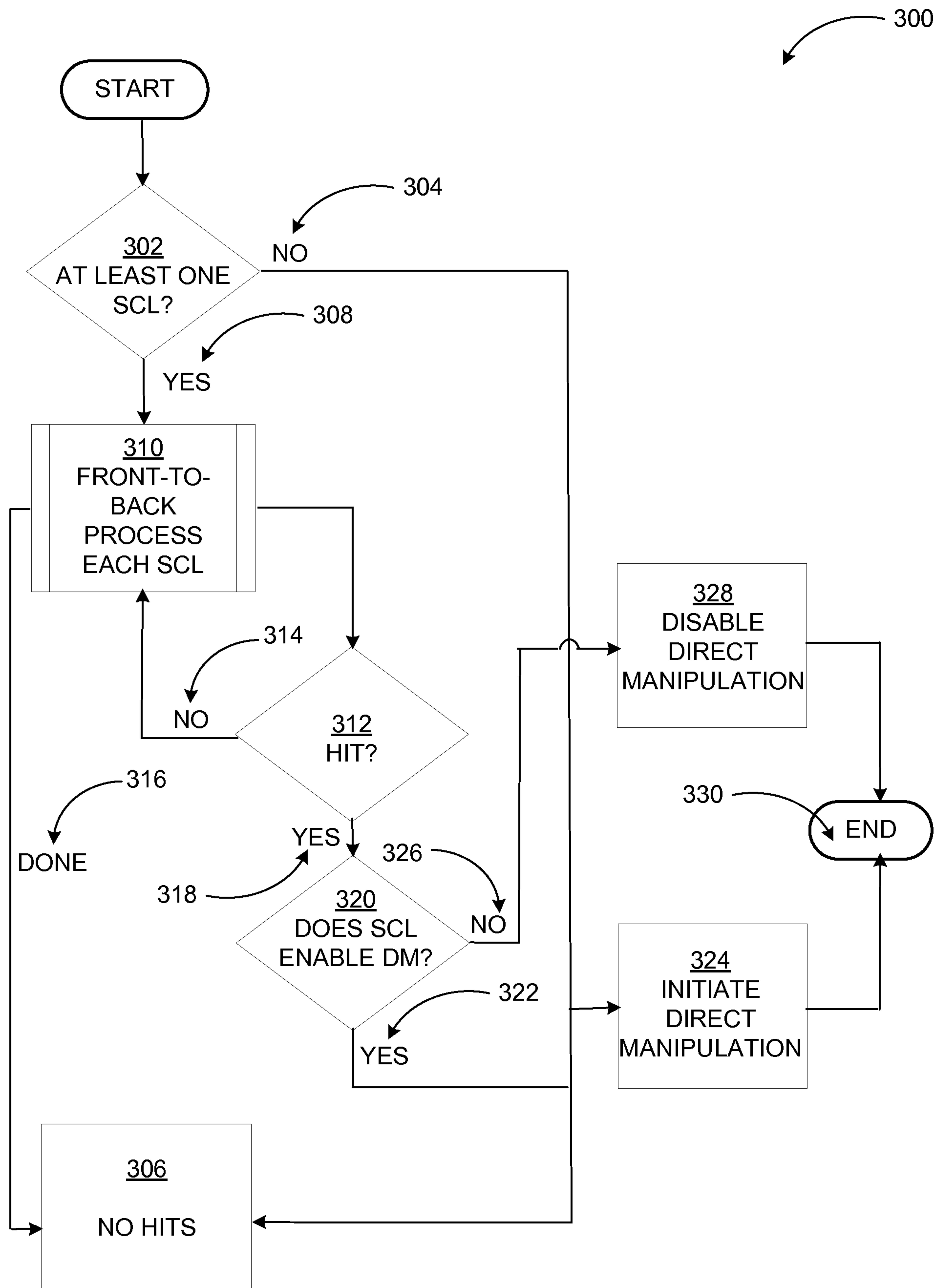
FIG. 3

4/5

**FIG. 4**

5/5

**FIG. 5**



**FIG. 3**