US 20180348783A1

(19) United States

(12) Patent Application Publication

Pitzer et al.

(10) Pub. No.: US 2018/0348783 A1

(43) Pub. Date: Dec. 6, 2018

(54) ASYNCHRONOUS IMAGE CLASSIFICATION

(71) Applicant: **Neato Robotics, Inc.**, Newark, CA (US)

(72) Inventors: **Charles Albert Pitzer**, Fremont, CA (US); **Griswald Brooks**, Fremont, CA (US); **Jose Capriles**, Foster City, CA (US); **Rachel Lucas**, Hayward, CA (US); **Kingman Yee**, Newark, CA (US)

(21) Appl. No.: **15/610,401**

(22) Filed: **May 31, 2017**

Publication Classification

(51) Int. Cl.

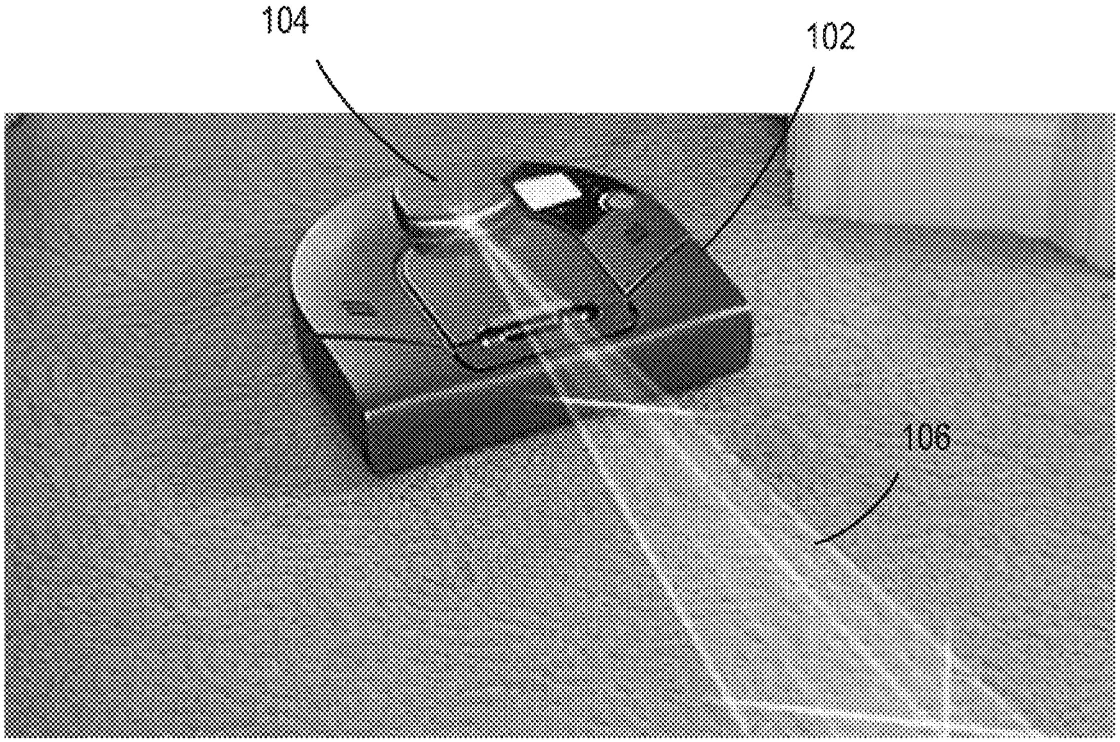
G05D 1/02	(2006.01)
G06K 9/00	(2006.01)
G06T 7/11	(2006.01)
H04N 5/38	(2006.01)
A47L 9/04	(2006.01)
A47L 9/28	(2006.01)
A47L 9/14	(2006.01)
A47L 9/00	(2006.01)

(52) U.S. Cl.

CPC ... ***G05D 1/0248*** (2013.01); *G05D 2201/0215* (2013.01); ***G06T 7/11*** (2017.01); ***H04N 5/38*** (2013.01); ***G06K 9/00684*** (2013.01); ***A47L 9/0477*** (2013.01); ***A47L 9/2857*** (2013.01); ***A47L 9/2852*** (2013.01); ***A47L 9/2847*** (2013.01); ***A47L 9/14*** (2013.01); ***A47L 9/2894*** (2013.01); ***A47L 9/2826*** (2013.01); ***A47L 9/2884*** (2013.01); ***A47L 9/2873*** (2013.01); ***A47L 9/2889*** (2013.01); ***G05D 1/0221*** (2013.01); ***G05D 1/0214*** (2013.01); ***G05D 1/0227*** (2013.01); ***G05D 1/024*** (2013.01); ***G05D 1/0285*** (2013.01); ***A47L 9/009*** (2013.01); ***G05D 1/0274*** (2013.01); *G06T 2207/20132* (2013.01); ***G06K 9/00671*** (2013.01)

(57) ABSTRACT

Embodiments provide methods and apparatus for asynchronously classifying images provided by a robot. In a reconnaissance/exploratory or first cleaning pass, unidentified objects are avoided. Images of the object are uploaded over the Internet to a remote object detection and classification system, and the location is indicated by the cleaning robot. When the remote system subsequently returns an object identification or classification, the object can be indicated as something to be avoided, or the cleaning robot can return to the location and clean over the object if it is determined not to be a hazard. In one embodiment, the object is passed over at reduced speed or with a cleaning brush turned off.



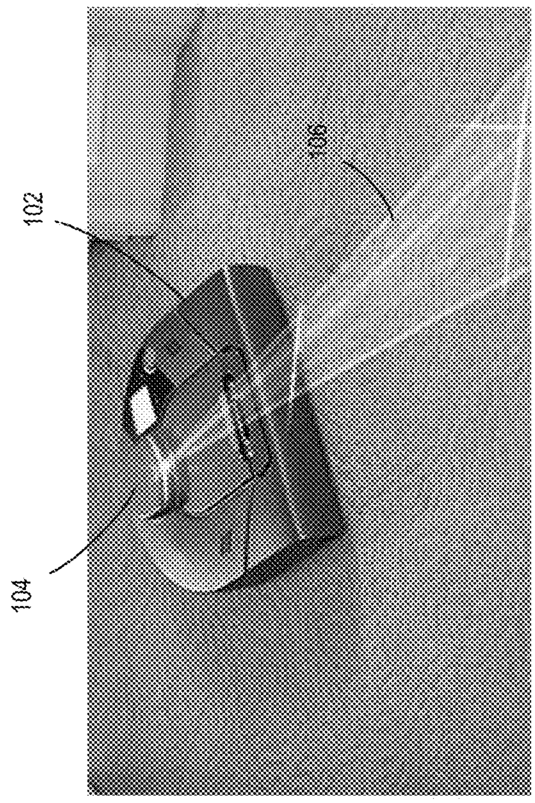


FIG. 1

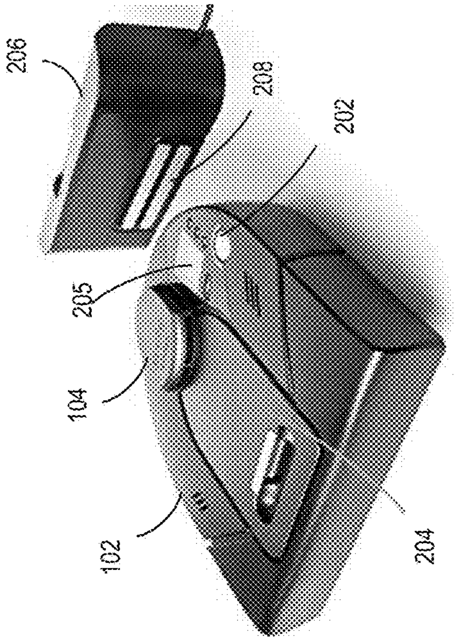


FIG. 2

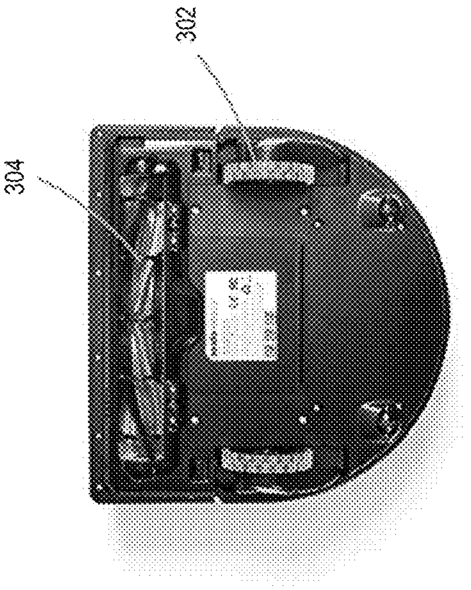


FIG. 3

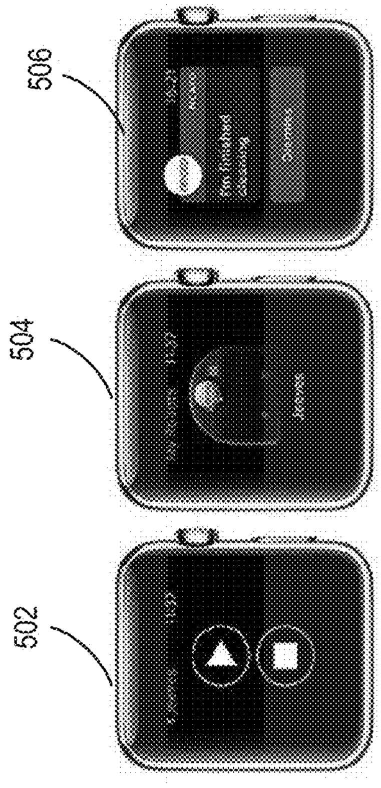


FIG. 5

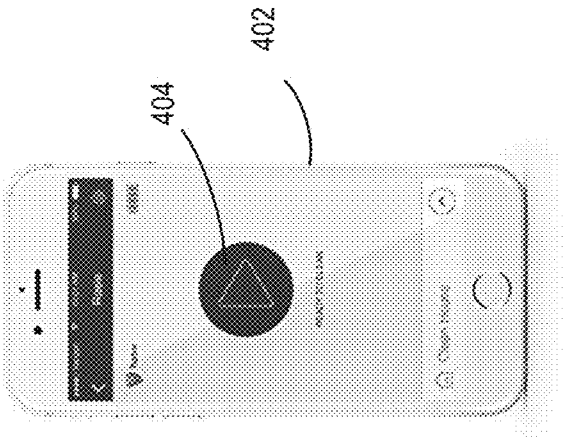


FIG. 4

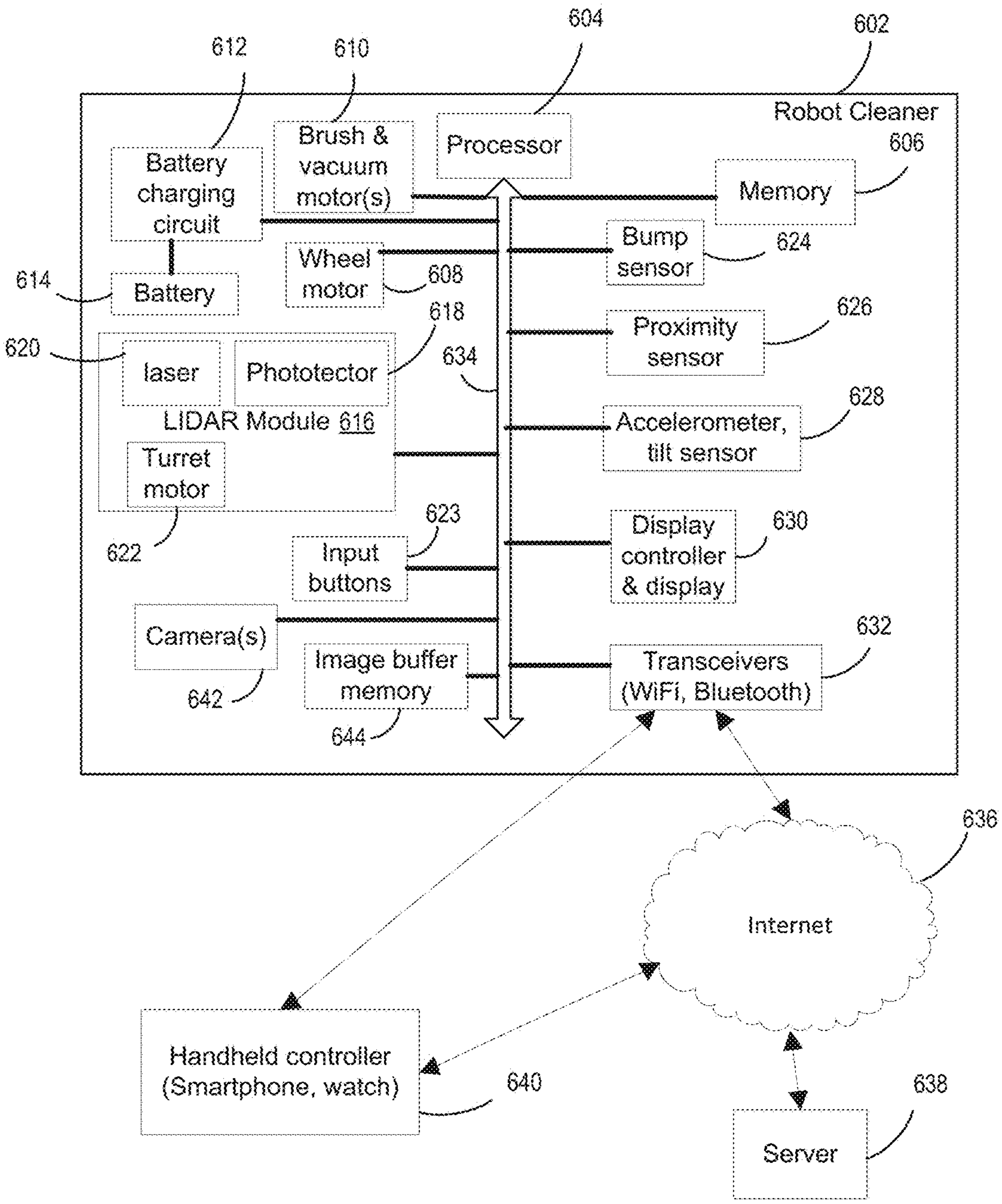

604
602
612
610
Robot Cleaner
Battery charging circuit
Brush & vacuum motor(s)
Processor
606
Memory
Bump sensor
624
Wheel motor
608
614
Battery
618
Proximity sensor
626
620
laser
Phototector
634
LIDAR Module 616
628
Accelerometer, tilt sensor
Turret motor
622
623
Input buttons
Display controller & display
630
Camera(s)
642
Image buffer memory
644
632
Transceivers (WiFi, Bluetooth)
636
Internet
Handheld controller (Smartphone, watch)
640
638
Server

FIG. 6

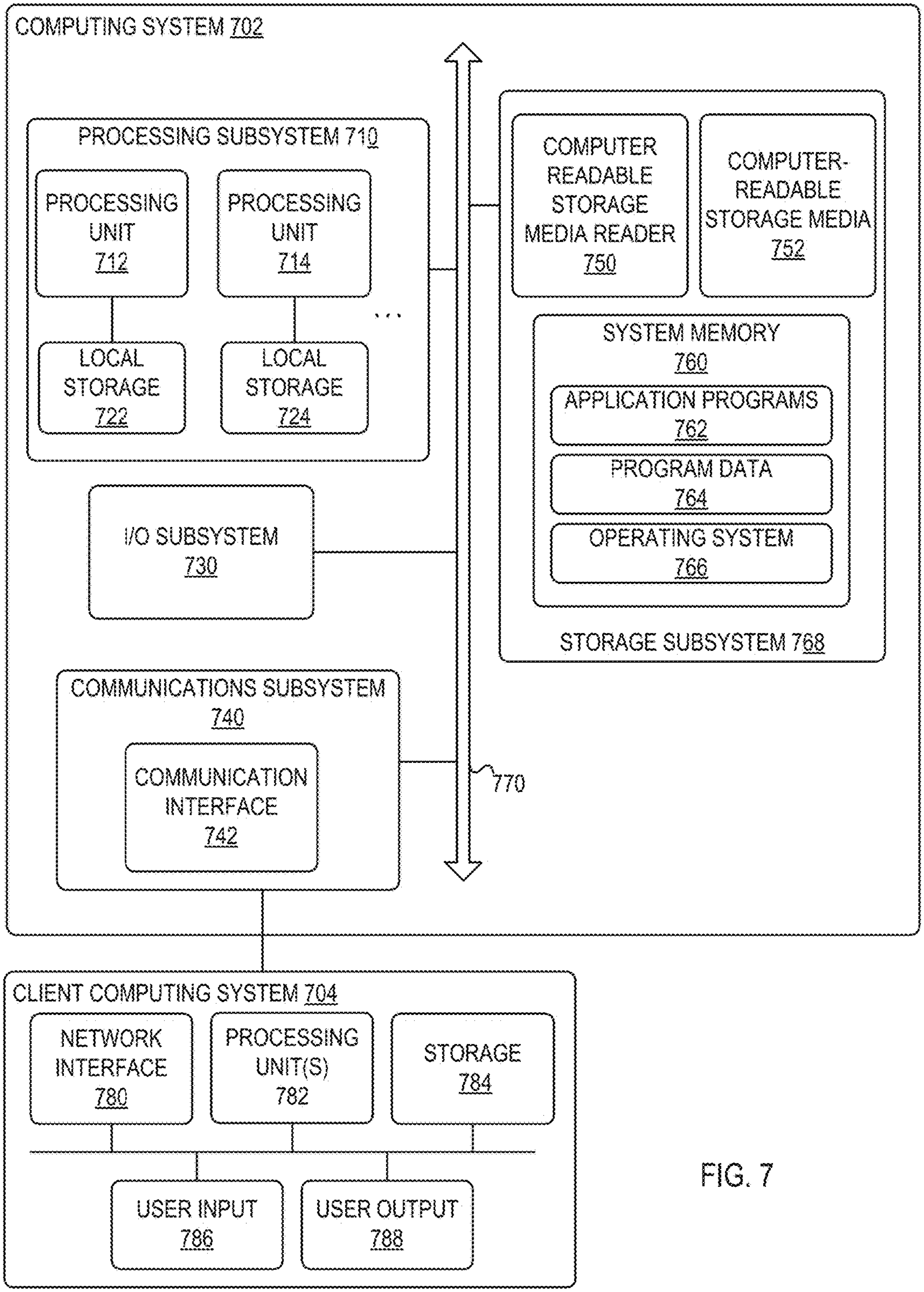

COMPUTING SYSTEM 702
PROCESSING SUBSYSTEM 710
PROCESSING UNIT 712
PROCESSING UNIT 714
LOCAL STORAGE 722
LOCAL STORAGE 724
...
I/O SUBSYSTEM 730
COMMUNICATIONS SUBSYSTEM 740
COMMUNICATION INTERFACE 742
COMPUTER READABLE STORAGE MEDIA READER 750
COMPUTER-READABLE STORAGE MEDIA 752
SYSTEM MEMORY 760
APPLICATION PROGRAMS 762
PROGRAM DATA 764
OPERATING SYSTEM 766
STORAGE SUBSYSTEM 768
770
CLIENT COMPUTING SYSTEM 704
NETWORK INTERFACE 780
PROCESSING UNIT(S) 782
STORAGE 784
USER INPUT 786
USER OUTPUT 788

FIG. 7

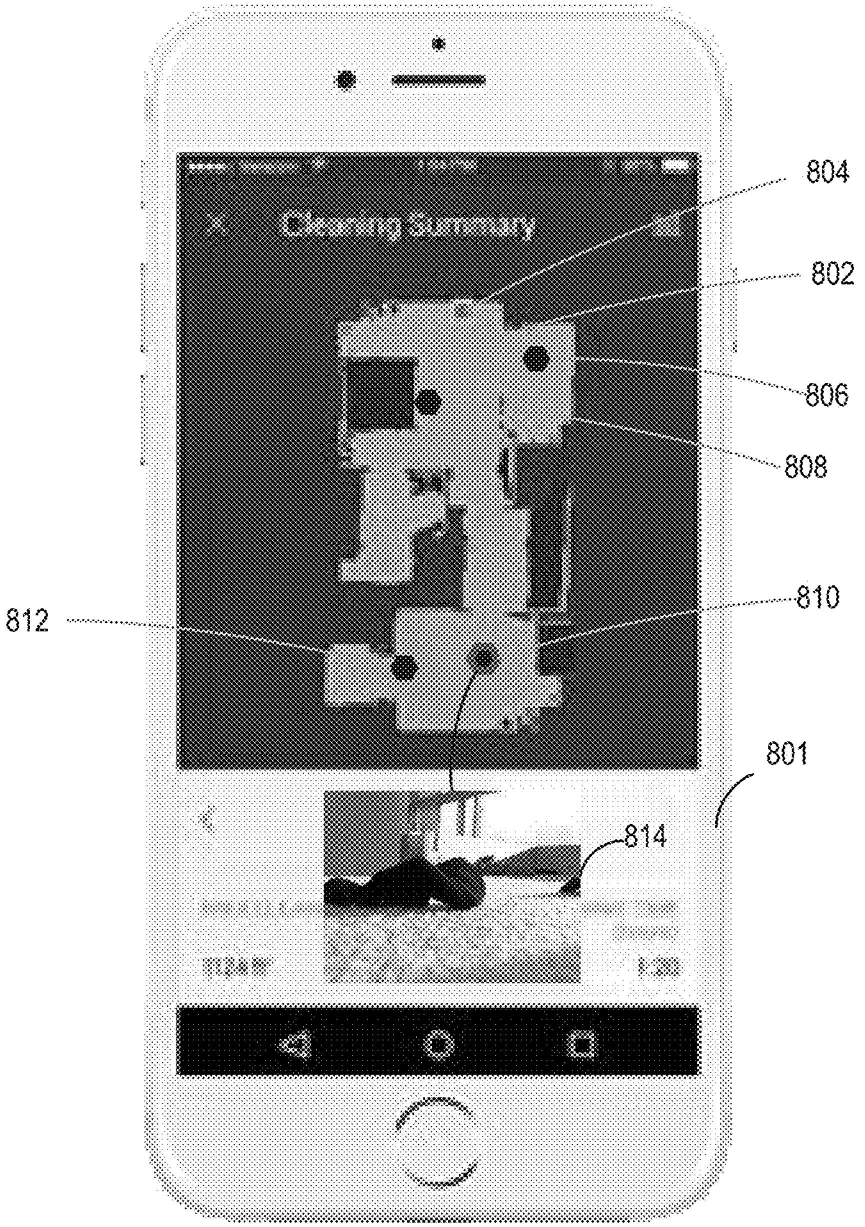


FIG. 8

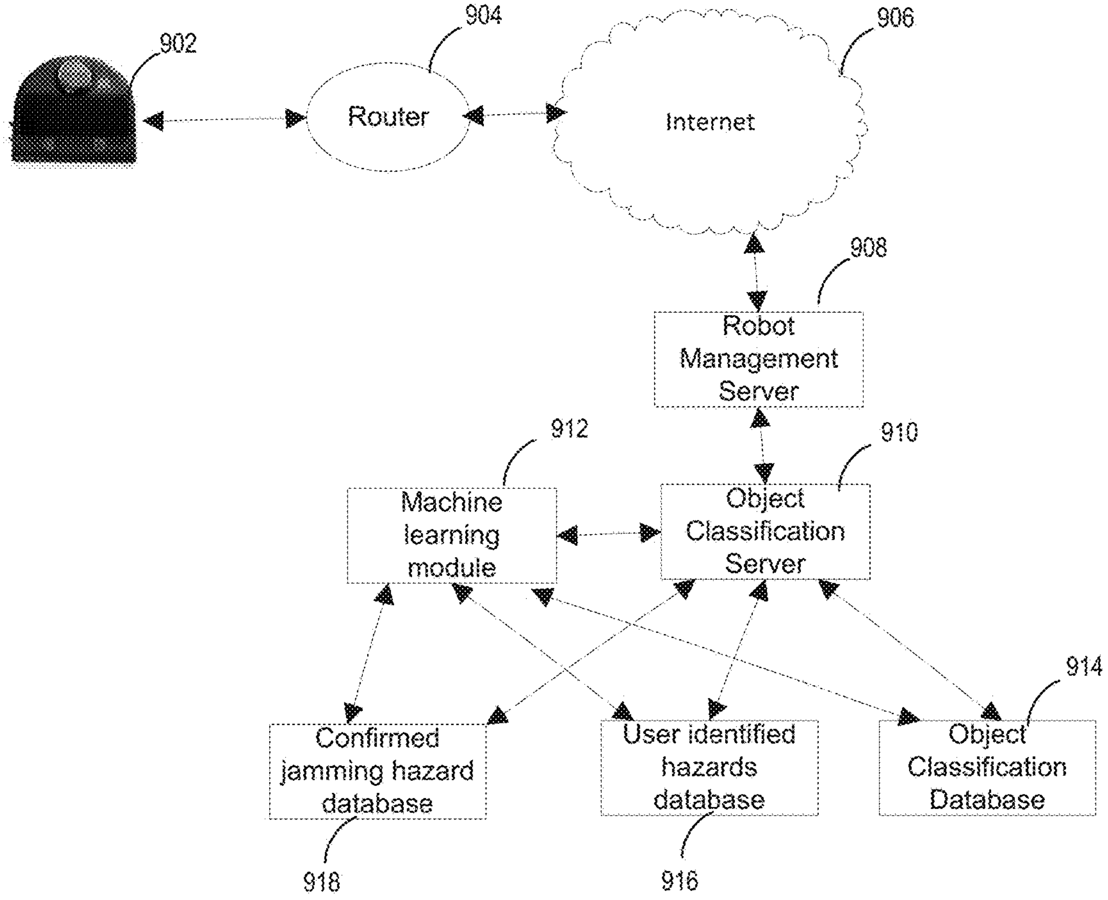

902
904
Router
906
Internet
908
Robot Management Server
912
Machine learning module
910
Object Classification Server
914
Confirmed jamming hazard database
User identified hazards database
Object Classification Database
918
916

FIG. 9

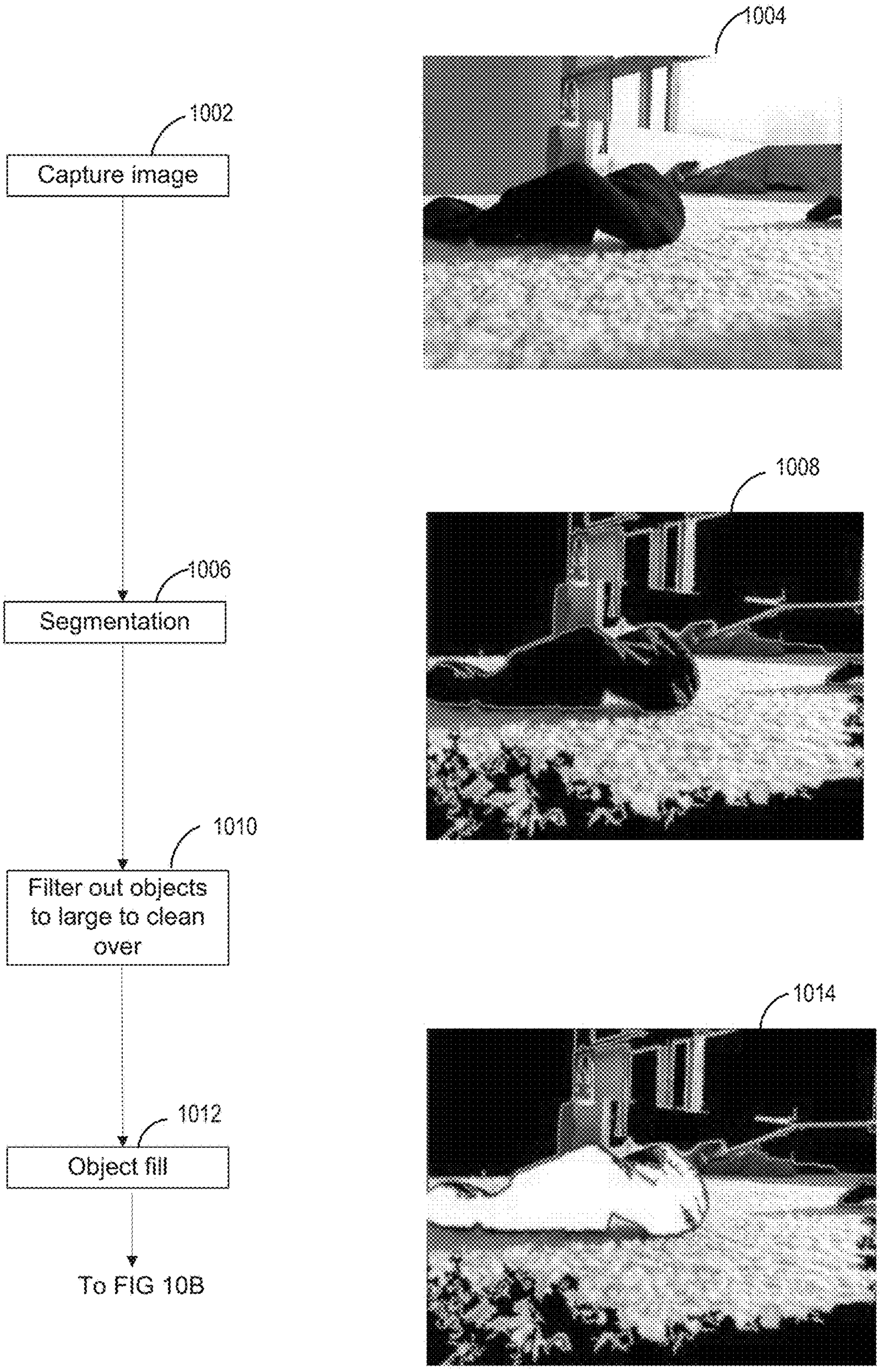

1004
1002
Capture image
1008
1006
Segmentation
1010
Filter out objects to large to clean over
1014
1012
Object fill
To FIG 10B

FIG. 10A

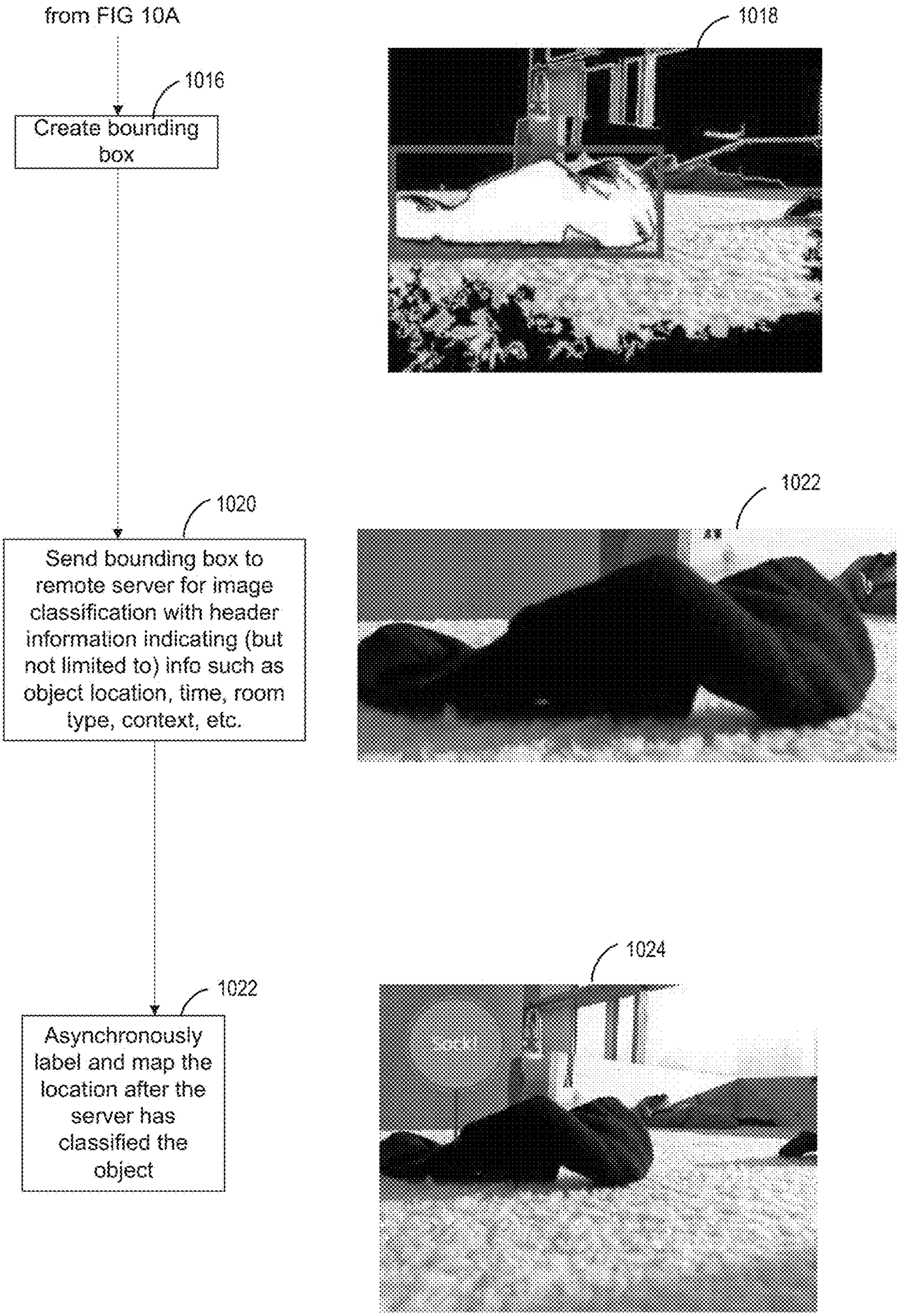

from FIG 10A
1016
Create bounding box
1018
1020
Send bounding box to remote server for image classification with header information indicating (but not limited to) info such as object location, time, room type, context, etc.
1022
1022
Asynchronously label and map the location after the server has classified the object
1024

FIG. 10B

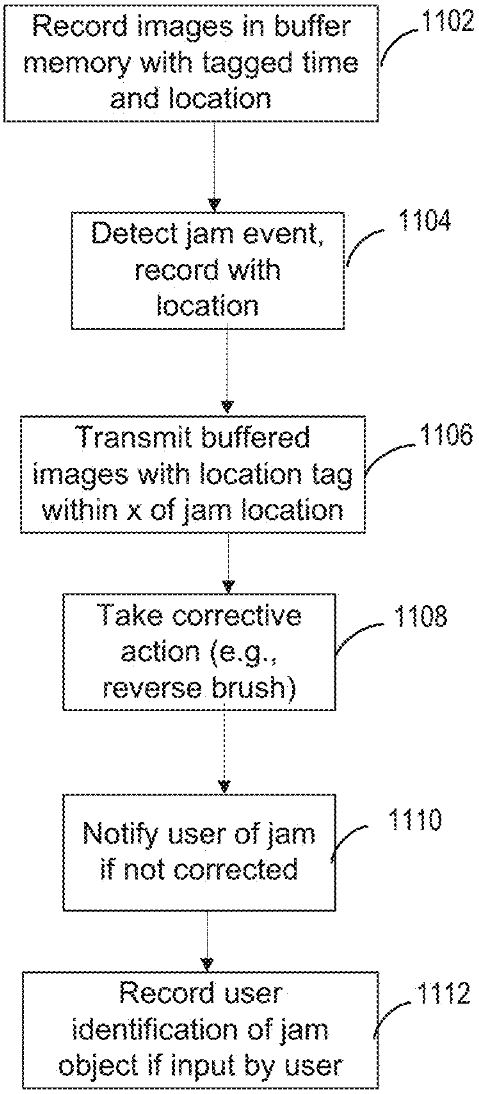

Record images in buffer memory with tagged time and location
1102
Detect jam event, record with location
1104
Transmit buffered images with location tag within x of jam location
1106
Take corrective action (e.g., reverse brush)
1108
Notify user of jam if not corrected
1110
Record user identification of jam object if input by user
1112

FIG. 11

ASYNCHRONOUS IMAGE CLASSIFICATION

BACKGROUND OF THE INVENTION

[0001] The present invention relates to robots which collect images, and in particular to cleaning robots with image detection capabilities.

[0002] Image detection and object classification has been used in many fields. There have been proposals to provide image recognition in household robots, such as cleaning robots. For example, US Pub. 20150289743 describes image recognition to determine a kind of dirt (e.g., hairs, spilled food) and select the appropriate cleaning capability.

[0003] US Pub. 20160167226 describes loading training data into the memory of a cleaning robot for use in machine learning for object recognition and classification, and also describes machine learning alternately implemented on remote servers over the Internet. The features may be identified using at least one of Scale-Invariant Feature Transform (SIFT) descriptors, Speeded Up Robust Features (SURF) descriptors, and Binary Robust Independent Elementary Features (BRIEF) descriptors. The classifier may be a Support Vector Machine. Classifier outputs can be utilized to determine appropriate behavior such as changing direction to avoid an obstacle. For example, information generated by classifiers can specify portions of a captured image that contain traversable floor and/or portions of the captured image that contain obstacles and/or non-traversable floor. The disclosures of the above publications are hereby incorporated herein by reference as providing background details on device elements and operations.

BRIEF SUMMARY OF THE INVENTION

[0004] Embodiments provide practical and economic methods and apparatus for asynchronously classifying images provided by a robot. In a reconnaissance/exploratory or first cleaning pass, unidentified objects are avoided. Images of the object are uploaded over the Internet to a remote object detection and classification system, and the location is indicated by the cleaning robot. When the remote system subsequently returns an object identification or classification, the object can be indicated as something to be avoided, or the cleaning robot can return to the location and clean over the object if it is determined not to be a hazard.

[0005] In one embodiment, when an event renders the cleaning robot stuck or jammed or otherwise inoperable, the images immediately prior to the event are examined for objects. Any objects, or simply the image, is tagged as a hazard. The tagged hazards are uploaded to a database of tagged hazards from other cleaning robots. New images and objects are compared to the tagged hazards database to identify hazards. This eliminates the need for determining the type of object with object recognition and classification—the robot will simply know that such an object is a hazard and has impacted other robots adversely.

[0006] In one embodiment, object images are provided to a user for the user to classify as a hazard or not a hazard. This can also be done asynchronously, with the robot returning to the object location, or not, depending upon a user response. Optionally, the user can also input a description of the object (e.g, sock). The user indications of objects as hazards can be uploaded with the images of the hazards as a tag to a hazard database.

[0007] In one embodiment, the cleaning robot can obtain additional information about the object. The additional information, or some additional information may be obtained upon the first encounter with the object. Alternately, the cleaning robot may return to the object when object classification is indefinite. The additional information can be additional image views of the object from different directions or angles. Multiple cameras on the cleaning robot can capture different angles, or the cleaning robot can be maneuvered around the object for different views. A bump or pressure sensor can be used with slight contact with the object to determine if it is hard or soft.

[0008] In one embodiment, a user completes a questionnaire and the answers are used to filter the potential object matches. For example, if the user does not have a pet, dog poop can be eliminated from the possible object classification. Conversely, if a user has a dog, dog poop can be added to the list of potential objects with a higher weighting of likelihood of a match. If a user has kids, toys can be weighted higher, or eliminated if a user doesn't have kids. Indicating birthdays can be used to increase the likelihood weighting of wrapping paper and ribbons around the time of the birthday.

[0009] In one embodiment, the object is passed over at reduced speed or with a cleaning brush turned off. A variety of other embodiments are described in the following drawings and description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. **1** is a diagram of a cleaning robot with a LIDAR turret according to an embodiment.

[0011] FIG. **2** is a diagram of a cleaning robot and charging station according to an embodiment.

[0012] FIG. **3** is a diagram of the underside of a cleaning robot according to an embodiment.

[0013] FIG. **4** is a diagram of a smartphone control application display for a cleaning robot according to an embodiment.

[0014] FIG. **5** is a diagram of a smart watch control application display for a cleaning robot according to an embodiment.

[0015] FIG. **6** is a diagram of a the electronic system for a cleaning robot according to an embodiment.

[0016] FIG. **7** is a simplified block diagram of a representative computing system and client computing system usable to implement certain embodiments of the present invention.

[0017] FIG. **8** is a diagram of an embodiment of a cleaning map indicating the locations of detected objects.

[0018] FIG. **9** is a diagram of an embodiment of a system for detecting hazards and classifying objects.

[0019] FIGS. **10**A-B are a diagram and flow chart of an embodiment of a method for detecting and classifying objects.

[0020] FIG. **11** is a flowchart of an embodiment of a method for detecting hazards and taking corrective action.

DETAILED DESCRIPTION OF THE INVENTION

Overall Architecture

[0021] FIG. **1** is a diagram of a cleaning robot with a LIDAR turret according to an embodiment. A cleaning robot **102** has a LIDAR (Light Detection and Ranging) turret **104**

which emits a rotating laser beam **106**. Detected reflections of the laser beam off objects are used to calculate both the distance to objects and the location of the cleaning robot. One embodiment of the distance calculation is set forth in U.S. Pat. No. 8,996,172, "Distance sensor system and method," the disclosure of which is incorporated herein by reference. The collected data is also used to create a map, using a SLAM (Simultaneous Location and Mapping) algorithm. One embodiment of a SLAM algorithm is described in U.S. Pat. No. 8,903,589, "Method and apparatus for simultaneous localization and mapping of mobile robot environment," the disclosure of which is incorporated herein by reference.

[0022] FIG. **2** is a diagram of a cleaning robot and charging station according to an embodiment. Cleaning robot **102** with turret **10** is shown. Also shown is a cover **204** which can be opened to access a dirt collection bag and the top side of a brush. Buttons **202** allow basic operations of the robot cleaner, such as starting a cleaning operation. A display **205** provides information to the user. Cleaning robot **102** can dock with a charging station **206**, and receive electricity through charging contacts **208**.

[0023] FIG. **3** is a diagram of the underside of a cleaning robot according to an embodiment. Wheels **302** move the cleaning robot, and a brush **304** helps free dirt to be vacuumed into the dirt bag.

[0024] FIG. **4** is a diagram of a smartphone control application display for a cleaning robot according to an embodiment. A smartphone **402** has an application that is downloaded to control the cleaning robot. An easy to use interface has a start button **404** to initiate cleaning.

[0025] FIG. **5** is a diagram of a smart watch control application display for a cleaning robot according to an embodiment. Example displays are shown. A display **502** provides and easy to use start button. A display **504** provides the ability to control multiple cleaning robots. A display **506** provides feedback to the user, such as a message that the cleaning robot has finished.

[0026] FIG. **6** is a high level diagram of a the electronic system for a cleaning robot according to an embodiment. A cleaning robot **602** includes a processor **604** that operates a program downloaded to memory **606**. The processor communicates with other components using a bus **634** or other electrical connections. In a cleaning mode, wheel motors **608** control the wheels independently to move and steer the robot. Brush and vacuum motors **610** clean the floor, and can be operated in different modes, such as a higher power intensive cleaning mode or a normal power mode.

[0027] LIDAR module **616** includes a laser **620** and a detector **616**. A turret motor **622** moves the laser and detector to detect objects up to 360 degrees around the cleaning robot. There are multiple rotations per second, such as about 5 rotations per second. Various sensors provide inputs to processor **604**, such as a bump sensor **624** indicating contact with an object, proximity sensor **626** indicating closeness to an object, and accelerometer and tilt sensors **628**, which indicate a drop-off (e.g., stairs) or a tilting of the cleaning robot (e.g., upon climbing over an obstacle). Examples of the usage of such sensors for navigation and other controls of the cleaning robot are set forth in U.S. Pat. No. 8,855,914, "Method and apparatus for traversing corners of a floored area with a robotic surface treatment apparatus," the disclosure of which is incorporated herein by reference. Other sensors may be included in other embodiments, such as a dirt sensor for detecting the amount of dirt being vacuumed, a motor current sensor for detecting when the motor is overloaded, such as due to being entangled in something, a floor sensor for detecting the type of floor, and an image sensor (camera) for providing images of the environment and objects.

[0028] A battery **614** provides power to the rest of the electronics though power connections (not shown). A battery charging circuit **612** provides charging current to battery **614** when the cleaning robot is docked with charging station **206** of FIG. **2**. Input buttons **623** allow control of robot cleaner **602** directly, in conjunction with a display **630**. Alternately, cleaning robot **602** may be controlled remotely, and send data to remote locations, through transceivers **632**.

[0029] Through the Internet **636**, and/or other network(s), the cleaning robot can be controlled, and can send information back to a remote user. A remote server **638** can provide commands, and can process data uploaded from the cleaning robot. A handheld smartphone or watch **640** can be operated by a user to send commands either directly to cleaning robot **602** (through Bluetooth, direct RF, a WiFi LAN, etc.) or can send commands through a connection to the internet **636**. The commands could be sent to server **638** for further processing, then forwarded in modified form to cleaning robot **602** over the internet **636**.

[0030] A camera or cameras **642** captures images of objects near the robot cleaner. In one embodiment, at least one camera is positioned to obtain images in front of the robot, showing where the robot is heading. The images are buffered in an image buffer memory **644**. The images may be video, or a series of still images. These images are stored for a certain period of time, such as 15 seconds-2 minutes, or up to 10 minutes, or for an entire cleaning operation between leaving a charging station and returning to the charging station. The images may subsequently be written over.

Computer Systems for Media Platform and Client System

[0031] Various operations described herein may be implemented on computer systems. FIG. **7** shows a simplified block diagram of a representative computing system **702** and client computing system **704** usable to implement certain embodiments of the present invention. In various embodiments, computing system **702** or similar systems may implement the cleaning robot processor system, remote server, or any other computing system described herein or portions thereof. Client computing system **704** or similar systems may implement user devices such as a smartphone or watch with a robot cleaner application.

[0032] Computing system **702** may be one of various types, including processor and memory, a handheld portable device (e.g., an iPhone® cellular phone, an iPad® computing tablet, a PDA), a wearable device (e.g., a Google Glass® head mounted display), a personal computer, a workstation, a mainframe, a kiosk, a server rack, or any other data processing system.

[0033] Computing system **702** may include processing subsystem **710**. Processing subsystem **710** may communicate with a number of peripheral systems via bus subsystem **770**. These peripheral systems may include I/O subsystem **730**, storage subsystem **768**, and communications subsystem **740**.

[0034] Bus subsystem **770** provides a mechanism for letting the various components and subsystems of server computing system **704** communicate with each other as intended. Although bus subsystem **770** is shown schematically as a single bus, alternative embodiments of the bus subsystem may utilize multiple buses. Bus subsystem **770** may form a local area network that supports communication in processing subsystem **710** and other components of server computing system **702**. Bus subsystem **770** may be implemented using various technologies including server racks, hubs, routers, etc. Bus subsystem **770** may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. For example, such architectures may include an Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus, which may be implemented as a Mezzanine bus manufactured to the IEEE P1386.1 standard, and the like.

[0035] I/O subsystem **730** may include devices and mechanisms for inputting information to computing system **702** and/or for outputting information from or via computing system **702**. In general, use of the term "input device" is intended to include all possible types of devices and mechanisms for inputting information to computing system **702**. User interface input devices may include, for example, a keyboard, pointing devices such as a mouse or trackball, a touchpad or touch screen incorporated into a display, a scroll wheel, a click wheel, a dial, a button, a switch, a keypad, audio input devices with voice command recognition systems, microphones, and other types of input devices. User interface input devices may also include motion sensing and/or gesture recognition devices such as the Microsoft Kinect® motion sensor that enables users to control and interact with an input device, the Microsoft Xbox® 360 game controller, devices that provide an interface for receiving input using gestures and spoken commands. User interface input devices may also include eye gesture recognition devices such as the Google Glass® blink detector that detects eye activity (e.g., "blinking" while taking pictures and/or making a menu selection) from users and transforms the eye gestures as input into an input device (e.g., Google Glass®). Additionally, user interface input devices may include voice recognition sensing devices that enable users to interact with voice recognition systems (e.g., Siri® navigator), through voice commands.

[0036] Other examples of user interface input devices include, without limitation, three dimensional (3D) mice, joysticks or pointing sticks, gamepads and graphic tablets, and audio/visual devices such as speakers, digital cameras, digital camcorders, portable media players, webcams, image scanners, fingerprint scanners, barcode reader 3D scanners, 3D printers, laser rangefinders, and eye gaze tracking devices. Additionally, user interface input devices may include, for example, medical imaging input devices such as computed tomography, magnetic resonance imaging, position emission tomography, medical ultrasonography devices. User interface input devices may also include, for example, audio input devices such as MIDI keyboards, digital musical instruments and the like.

[0037] User interface output devices may include a display subsystem, indicator lights, or non-visual displays such as audio output devices, etc. The display subsystem may be a cathode ray tube (CRT), a flat-panel device, such as that using a liquid crystal display (LCD) or plasma display, a projection device, a touch screen, and the like. In general, use of the term "output device" is intended to include all possible types of devices and mechanisms for outputting information from computing system **702** to a user or other computer. For example, user interface output devices may include, without limitation, a variety of display devices that visually convey text, graphics and audio/video information such as monitors, printers, speakers, headphones, automotive navigation systems, plotters, voice output devices, and modems.

[0038] Processing subsystem **710** controls the operation of computing system **702** and may comprise one or more processing units **712**, **714**, etc. A processing unit may include one or more processors, including single core processor or multicore processors, one or more cores of processors, or combinations thereof. In some embodiments, processing subsystem **710** may include one or more special purpose co-processors such as graphics processors, digital signal processors (DSPs), or the like. In some embodiments, some or all of the processing units of processing subsystem **710** may be implemented using customized circuits, such as application specific integrated circuits (ASICs), or field programmable gate arrays (FPGAs). In some embodiments, such integrated circuits execute instructions that are stored on the circuit itself. In other embodiments, processing unit(s) may execute instructions stored in local storage, e.g., local storage **722**, **724**. Any type of processors in any combination may be included in processing unit(s) **712**, **714**.

[0039] In some embodiments, processing subsystem **710** may be implemented in a modular design that incorporates any number of modules (e.g., blades in a blade server implementation). Each module may include processing unit (s) and local storage. For example, processing subsystem **710** may include processing unit **712** and corresponding local storage **722**, and processing unit **714** and corresponding local storage **724**.

[0040] Local storage **722**, **724** may include volatile storage media (e.g., conventional DRAM, SRAM, SDRAM, or the like) and/or non-volatile storage media (e.g., magnetic or optical disk, flash memory, or the like). Storage media incorporated in local storage **722**, **724** may be fixed, removable or upgradeable as desired. Local storage **722**, **724** may be physically or logically divided into various subunits such as a system memory, a ROM, and a permanent storage device. The system memory may be a read-and-write memory device or a volatile read-and-write memory, such as dynamic random access memory. The system memory may store some or all of the instructions and data that processing unit(s) **712**, **714** need at runtime. The ROM may store static data and instructions that are needed by processing unit(s) **712**, **714**. The permanent storage device may be a non-volatile read-and-write memory device that may store instructions and data even when a module including one or more processing units **712**, **714** and local storage **722**, **724** is powered down. The term "storage medium" as used herein includes any medium in which data may be stored indefinitely (subject to overwriting, electrical disturbance, power loss, or the like) and does not include carrier waves and transitory electronic signals propagating wirelessly or over wired connections.

[0041] In some embodiments, local storage **722**, **724** may store one or more software programs to be executed by processing unit(s) **712**, **714**, such as an operating system and/or programs implementing various server functions such as functions of UPP system **102**, or any other server(s) associated with UPP system **102**. "Software" refers generally to sequences of instructions that, when executed by processing unit(s) **712**, **714** cause computing system **702** (or portions thereof) to perform various operations, thus defining one or more specific machine implementations that execute and perform the operations of the software programs. The instructions may be stored as firmware residing in read-only memory and/or program code stored in non-volatile storage media that may be read into volatile working memory for execution by processing unit(s) **712**, **714**. In some embodiments the instructions may be stored by storage subsystem **768** (e.g., computer readable storage media). In various embodiments, the processing units may execute a variety of programs or code instructions and may maintain multiple concurrently executing programs or processes. At any given time, some or all of the program code to be executed may be resident in local storage **722**, **724** and/or in storage subsystem including potentially on one or more storage devices. Software may be implemented as a single program or a collection of separate programs or program modules that interact as desired. From local storage **722**, **724** (or non-local storage described below), processing unit(s) **712**, **714** may retrieve program instructions to execute and data to process in order to execute various operations described above.

[0042] Storage subsystem **768** provides a repository or data store for storing information that is used by computing system **702**. Storage subsystem **768** provides a tangible non-transitory computer-readable storage medium for storing the basic programming and data constructs that provide the functionality of some embodiments. Software (programs, code modules, instructions) that when executed by processing subsystem **710** provide the functionality described above may be stored in storage subsystem **768**. The software may be executed by one or more processing units of processing subsystem **710**. Storage subsystem **768** may also provide a repository for storing data used in accordance with the present invention.

[0043] Storage subsystem **768** may include one or more non-transitory memory devices, including volatile and non-volatile memory devices. As shown in FIG. **7**, storage subsystem **768** includes a system memory **760** and a computer-readable storage media **752**. System memory **760** may include a number of memories including a volatile main RAM for storage of instructions and data during program execution and a non-volatile ROM or flash memory in which fixed instructions are stored. In some implementations, a basic input/output system (BIOS), containing the basic routines that help to transfer information between elements within computing system **702**, such as during start-up, may typically be stored in the ROM. The RAM typically contains data and/or program modules that are presently being operated and executed by processing subsystem **710**. In some implementations, system memory **760** may include multiple different types of memory, such as static random access memory (SRAM) or dynamic random access memory (DRAM). Storage subsystem **768** may be based on magnetic, optical, semiconductor, or other data storage media. Direct attached storage, storage area networks, network-attached storage, and the like may be used. Any data stores or other collections of data described herein as being produced, consumed, or maintained by a service or server may be stored in storage subsystem **768**.

[0044] By way of example, and not limitation, as depicted in FIG. **7**, system memory **760** may store application programs **762**, which may include client applications, Web browsers, mid-tier applications, relational database management systems (RDBMS), etc., program data **764**, and one or more operating systems **766**. By way of example, an example operating systems may include various versions of Microsoft Windows®, Apple Macintosh®, and/or Linux operating systems, a variety of commercially-available UNIX® or UNIX-like operating systems (including without limitation the variety of GNU/Linux operating systems, the Google Chrome® OS, and the like) and/or mobile operating systems such as iOS, Windows® Phone, Android® OS, BlackBerry® 10 OS, and Palm® OS operating systems.

[0045] Computer-readable storage media **752** may store programming and data constructs that provide the functionality of some embodiments. Software (programs, code modules, instructions) that when executed by processing subsystem **710** a processor provide the functionality described above may be stored in storage subsystem **768**. By way of example, computer-readable storage media **752** may include non-volatile memory such as a hard disk drive, a magnetic disk drive, an optical disk drive such as a CD ROM, DVD, a Blu-Ray® disk, or other optical media. Computer-readable storage media **752** may include, but is not limited to, Zip® drives, flash memory cards, universal serial bus (USB) flash drives, secure digital (SD) cards, DVD disks, digital video tape, and the like. Computer-readable storage media **752** may also include, solid-state drives (SSD) based on non-volatile memory such as flash-memory based SSDs, enterprise flash drives, solid state ROM, and the like, SSDs based on volatile memory such as solid state RAM, dynamic RAM, static RAM, DRAM-based SSDs, magnetoresistive RAM (MRAM) SSDs, and hybrid SSDs that use a combination of DRAM and flash memory based SSDs. Computer-readable media **752** may provide storage of computer-readable instructions, data structures, program modules, and other data for computing system **702**.

[0046] In certain embodiments, storage subsystem **768** may also include a computer-readable storage media reader **750** that may further be connected to computer-readable storage media **752**. Together and, optionally, in combination with system memory **760**, computer-readable storage media **752** may comprehensively represent remote, local, fixed, and/or removable storage devices plus storage media for storing computer-readable information.

[0047] In certain embodiments, computing system **702** may provide support for executing one or more virtual machines. Computing system **702** may execute a program such as a hypervisor for facilitating the configuring and managing of the virtual machines. Each virtual machine may be allocated memory, compute (e.g., processors, cores), I/O, and networking resources. Each virtual machine typically runs its own operating system, which may be the same as or different from the operating systems executed by other virtual machines executed by computing system **702**. Accordingly, multiple operating systems may potentially be run concurrently by computing system **702**. Each virtual machine generally runs independently of the other virtual machines.

[0048] Communication subsystem **740** provides an interface to other computer systems and networks. Communication subsystem **740** serves as an interface for receiving data from and transmitting data to other systems from computing system **702**. For example, communication subsystem **740** may enable computing system **702** to establish a communication channel to one or more client computing devices via the Internet for receiving and sending information from and to the client computing devices.

[0049] Communication subsystem **740** may support both wired and/or wireless communication protocols. For example, in certain embodiments, communication subsystem **740** may include radio frequency (RF) transceiver components for accessing wireless voice and/or data networks (e.g., using cellular telephone technology, advanced data network technology, such as 3G, 4G or EDGE (enhanced data rates for global evolution), WiFi (IEEE 802.11 family standards, or other mobile communication technologies, or any combination thereof), global positioning system (GPS) receiver components, and/or other components. In some embodiments communication subsystem **740** may provide wired network connectivity (e.g., Ethernet) in addition to or instead of a wireless interface.

[0050] Communication subsystem **740** may receive and transmit data in various forms. For example, in some embodiments, communication subsystem **740** may receive input communication in the form of structured and/or unstructured data feeds, event streams, event updates, and the like. For example, communication subsystem **740** may be configured to receive (or send) data feeds in real-time from users of social media networks and/or other communication services such as Twitter® feeds, Facebook® updates, web feeds such as Rich Site Summary (RSS) feeds, and/or real-time updates from one or more third party information sources.

[0051] In certain embodiments, communication subsystem **740** may be configured to receive data in the form of continuous data streams, which may include event streams of real-time events and/or event updates, that may be continuous or unbounded in nature with no explicit end. Examples of applications that generate continuous data may include, for example, sensor data applications, financial tickers, network performance measuring tools (e.g. network monitoring and traffic management applications), clickstream analysis tools, automobile traffic monitoring, and the like.

[0052] Communication subsystem **740** may also be configured to output the structured and/or unstructured data feeds, event streams, event updates, and the like to one or more databases that may be in communication with one or more streaming data source computers coupled to computing system **702**.

[0053] Communication subsystem **740** may provide a communication interface **742**, e.g., a WAN interface, which may provide data communication capability between the local area network (bus subsystem **770**) and a larger network, such as the Internet. Conventional or other communications technologies may be used, including wired (e.g., Ethernet, IEEE 802.3 standards) and/or wireless technologies (e.g., Wi-Fi, IEEE 802.11 standards).

[0054] Computing system **702** may operate in response to requests received via communication interface **742**. Further, in some embodiments, communication interface **742** may connect computing systems **702** to each other, providing scalable systems capable of managing high volumes of activity. Conventional or other techniques for managing server systems and server farms (collections of server systems that cooperate) may be used, including dynamic resource allocation and reallocation.

[0055] Computing system **702** may interact with various user-owned or user-operated devices via a wide-area network such as the Internet. An example of a user-operated device is shown in FIG. **7** as client computing system **702**. Client computing system **704** may be implemented, for example, as a consumer device such as a smart phone, other mobile phone, tablet computer, wearable computing device (e.g., smart watch, eyeglasses), desktop computer, laptop computer, and so on.

[0056] For example, client computing system **704** may communicate with computing system **702** via communication interface **742**. Client computing system **704** may include conventional computer components such as processing unit(s) **782**, storage device **784**, network interface **780**, user input device **786**, and user output device **788**. Client computing system **704** may be a computing device implemented in a variety of form factors, such as a desktop computer, laptop computer, tablet computer, smart phone, other mobile computing device, wearable computing device, or the like.

[0057] Processing unit(s) **782** and storage device **784** may be similar to processing unit(s) **712**, **714** and local storage **722**, **724** described above. Suitable devices may be selected based on the demands to be placed on client computing system **704**; for example, client computing system **704** may be implemented as a "thin" client with limited processing capability or as a high-powered computing device. Client computing system **704** may be provisioned with program code executable by processing unit(s) **782** to enable various interactions with computing system **702** of a message management service such as accessing messages, performing actions on messages, and other interactions described above. Some client computing systems **704** may also interact with a messaging service independently of the message management service.

[0058] Network interface **780** may provide a connection to a wide area network (e.g., the Internet) to which communication interface **740** of computing system **702** is also connected. In various embodiments, network interface **780** may include a wired interface (e.g., Ethernet) and/or a wireless interface implementing various RF data communication standards such as Wi-Fi, Bluetooth, or cellular data network standards (e.g., 3G, 4G, LTE, etc.).

[0059] User input device **786** may include any device (or devices) via which a user may provide signals to client computing system **704**; client computing system **704** may interpret the signals as indicative of particular user requests or information. In various embodiments, user input device **786** may include any or all of a keyboard, touch pad, touch screen, mouse or other pointing device, scroll wheel, click wheel, dial, button, switch, keypad, microphone, and so on.

[0060] User output device **788** may include any device via which client computing system **704** may provide information to a user. For example, user output device **788** may include a display to display images generated by or delivered to client computing system **704**. The display may incorporate various image generation technologies, e.g., a liquid crystal display (LCD), light-emitting diode (LED) including organic light-emitting diodes (OLED), projection system, cathode ray tube (CRT), or the like, together with supporting electronics (e.g., digital-to-analog or analog-to-digital converters, signal processors, or the like). Some embodiments may include a device such as a touchscreen that function as both input and output device. In some embodiments, other user output devices **788** may be provided in addition to or instead of a display. Examples include indicator lights, speakers, tactile "display" devices, printers, and so on.

[0061] Some embodiments include electronic components, such as microprocessors, storage and memory that store computer program instructions in a computer readable storage medium. Many of the features described in this specification may be implemented as processes that are specified as a set of program instructions encoded on a computer readable storage medium. When these program instructions are executed by one or more processing units, they cause the processing unit(s) to perform various operation indicated in the program instructions. Examples of program instructions or computer code include machine code, such as is produced by a compiler, and files including higher-level code that are executed by a computer, an electronic component, or a microprocessor using an interpreter. Through suitable programming, processing unit(s) **712**, **714** and **782** may provide various functionality for computing system **702** and client computing system **704**, including any of the functionality described herein as being performed by a server or client, or other functionality associated with message management services.

[0062] It will be appreciated that computing system **702** and client computing system **704** are illustrative and that variations and modifications are possible. Computer systems used in connection with embodiments of the present invention may have other capabilities not specifically described here. Further, while computing system **702** and client computing system **704** are described with reference to particular blocks, it is to be understood that these blocks are defined for convenience of description and are not intended to imply a particular physical arrangement of component parts. For instance, different blocks may be but need not be located in the same facility, in the same server rack, or on the same motherboard. Further, the blocks need not correspond to physically distinct components. Blocks may be configured to perform various operations, e.g., by programming a processor or providing appropriate control circuitry, and various blocks might or might not be reconfigurable depending on how the initial configuration is obtained. Embodiments of the present invention may be realized in a variety of apparatus including electronic devices implemented using any combination of circuitry and software.

Asynchronous Classification of Objects in Images

[0063] In one embodiment, the cleaning robot includes a camera, and can upload pictures through a Wireless Local Area Network (WLAN) and the Internet to a server or other computer that performs object recognition.

[0064] FIG. **8** is a diagram of an embodiment of a cleaning map created by a cleaning robot. A smartphone **801** (or tablet or other display device) shows a cleaning area or map **802** that has been mapped. A location of a robot charging station **804** is indicated. Also indicated are objects detected as the robot moved around, with the location on map **802** of objects indicated by icons **806**, **808**, **810** and **812**. A user can touch an icon, such as icon **810**, which is highlighted when touched. An image **814** of the object at the location of icon **810** is then displayed. The user can indicate the object is a potential hazard by swiping the image to the right, or indicate that it is not a hazard by swiping to the left. Alternately, other methods of displaying and selecting can be used, such as a list with yes and no buttons.

[0065] FIG. **9** is a diagram of an embodiment of an asynchronous object classification system. A cleaning robot **902** uploads images, or portions of images, by WiFi to a router **904**, which is connected to Internet **906**. The images are provided to a robot management server **908**, which can communicate with an application on a user device, such as a smartphone as shown in FIG. **8**. Server **912** also communicates the images to an Object Classification Server **910**. A machine learning module **912** can be invoked on server **910**, or on a separate server. Three databases are shown, although they could be combined in a single database, or further segmented. An Object Classification Database **914** is used to store images from robots and training images for machine learning module **912**. Database **914** is invoked to classify and identify objects in submitted videos.

[0066] User identified hazards database **916** stores images that have been manually identified as hazards by users, such as by the mechanism described in FIG. **8**. This can either be used in conjunction with database **914**, or separately. In one embodiment, the images are not classified or identified at all. Rather, if a submitted image is a near match to something identified as a hazard in database **916**, a response to the submitting robot is a probability (90%, 80%, 70%, 50%, etc.) that the object in the image is a hazard. The robot will then avoid the hazard, unless overruled by the user as described further below.

[0067] Confirmed jamming hazard database **918** stores images taken just before a robot became jammed or otherwise rendered inoperable. Again, if a submitted image is a near match to something identified as a hazard in database **918**, a response to the submitting robot is a probability (90%, 80%, 70%, 50%, etc.) that the object in the image is a hazard. The probability indicates the degree of confidence that the object in the submitted image is the same or similar to a confirmed hazard object in database **918**. The robot will then avoid the hazard, unless overruled by the user.

[0068] Object classification database **918**, in one embodiment, includes tags for each object indicating whether they are a hazard, and a degree of confidence that they are a hazard. If a submitted image is a near match to something identified as a hazard in object classification database **914**, a response to the submitting robot is a probability that the object in the image is a hazard. This response may be instead of, or in addition to, providing an object classification and/or object identification, along with a degree of confidence in the classification and/or identification.

[0069] In one embodiment, submitted messages are compared to images in all three databases, and the response is a weighted combination of the matches from the three databases. In one embodiment, matches from the confirmed jamming hazard database are weighted highest, then matches from the user identified hazard database, then matches from the object classification database.

[0070] In one embodiment, as described above, objects are identified as hazards with a percentage probability. Such hazards are then maintained as marked on a map of the environment and are avoided. The default setting for the probability that an object is a hazard is set to being above a default percentage, such as 10-30%. The default setting could be changed manually by a user, or could be part of a cleaning style, such as set forth in co-pending U.S. patent application Ser. No. 15/475,983, filed Mar. 31, 2017, entitled “ROBOT WITH AUTOMATIC STYLES,” the disclosure of which is hereby incorporated herein by reference. For example, a “fast” style would automatically set the threshold low, such as somewhere in the range 10-20%, while the “thorough” style would set the threshold higher, to try to clean more potential hazards, such as somewhere in the range 30-50%.

Turning Off Robot Brush

[0071] In one embodiment, the robot has a cleaning brush and performs its normal cleaning operation, except that it proceeds with caution over (and around) unknown objects by turning off the brush or reducing the speed. This avoids the primary mode of entanglements. After the object is classified, compared to the user identified database, etc., the robot can return the those areas for a “touch up” cleaning with the brush turned on, if it is safe. In one embodiment, the robot will run over unidentified objects before classification, with the brush off, only if they are detected to be sufficiently small. Sufficiently small may indicated that the robot can pass over the object without contact. Alternately, if the object has been determined to be soft and compressible, an object that will partially contact the brush may be run over.

User Identification of Hazards

[0072] In one embodiment, the hazards with a certainty less than a threshold, such as 80-90%, are presented to, or made available to, the user. The display can be the image, such as image **814** in FIG. **8**, with or without the probability. The user can then indicate whether the object is indeed a hazard or not. Again, the user can change the default settings to require a higher or lower percentage probability of the object being a hazard before it is presented to the user. The user indications are then added to the user identified hazard database **916**. The images can be sent to robot management server **908** and then relayed to the user. Alternately, the images can be stored for the use to view the next time the user accesses the application on the user device. A text or other notification can be sent to the user to prompt review in real time. The user can adjust the settings to enable or disable such a notification. Optionally, the user can also input a description of the object (e.g, sock). The user indications of objects as hazards can be uploaded with the images of the hazards as a tag to a hazard database and also an object identification database. A user can elect whether to be prompted to identify hazards at all, or not to be bothered. Incentives may be offered to the user to identify hazards, such as a discount on future purchases. The user may simply indicate whether it is a hazard or not, such as by a right or left swipe, clicking a yes/no button, doing a tap or double tap or X or other gesture, etc. The user can also be prompted to type in an identification, or select from a list of potential matches identified by the remote object classification server.

[0073] In one embodiment, user identified hazard database **916** contains not only images identified as hazards by a user, but also images identified as not being a hazard. Thus, a submitted image can be compared to both. If the image is more similar to a non-hazard image than a hazard image, it can be indicated to have a low probability of being a hazard. Similarly, confirmed jamming hazard database **918** may also contain images of objects that turned out not to be jamming hazards. This can be images that jammed a robot, but the robot was able to unjam through reversing the brush. This can also be images where an object is detected, but the robot moves over and cleans the object, with no jam occurring. Again, newly submitted images can be compared to both confirmed jamming hazards and confirmed non-hazards. It should be noted that although jamming is described as an example, any other action that renders the robot inoperable or partially inoperable is also covered by jamming, such as requiring increased power due to partial clogging of the robot or the robot getting stuck and unable to move, or trapped in a small area.

Object Identification Process

[0074] FIGS. **10**A-B are a diagram and flow chart of an embodiment of a method for detecting and classifying objects. In a step **1002**, an image is captured, such as image **1004**, by a camera on the robot. The image can be initially processed through cropping in the processor of the robot, or can simply be sent to a remote server for processing. Alternately, the image can be sent to an application on a user device for processing. The image can be still images taken at a periodic time, or taken at different locations, such as every 6 inches or 1-2 feet.

[0075] In step **1006**, segmentation of the image is performed, to produce the lines shown in image **1008**. Objects are identified by looking for long lines that enclose an area, or simply the longest continuous line. The area between those lines that enclose the area are filled (**1012**), such as shown in the example of image **1014**. Objects that are too large for the robot to pass over are filtered out (**1010**). For example, furniture, walls, etc. will be filtered out because the size can be determined from the segmentation and object fill. In addition, the robot LIDAR can determine the distance of the potential object from the robot, and from that and image analysis of the filled object, can determine its size. In one embodiment, 3D LIDAR can be used to estimate the object size.

[0076] In FIG. **10**B, which continues FIG. **10**A, a bounding box is created (**1016**), such as the example shown in image **1018**. This is used to crop the image to just contain the object, such as shown in image **1022**. The portion of the image in the bounding box is then sent to the remote server for image classification (**1020**). The image will be accompanied by a header of information indicating (but not limited to) information such as object location, time, room type, context, etc. The remote server will then classify the object as described above, and asynchronously return the classification information to the robot (**1022**). The robot will store the classification label, and may include the label on an image that may be presented to the user on a user device, such as shown in image **1024**.

[0077] In practice, the same object may occur in different images as the robot approaches or goes by the object. In one embodiment, the object classification server does image matching, in combination with analyzing the location data tagged with the images, to determine if the same object is indicated in multiple images. The best image of the object is then returned to the robot. The image may also, or instead, be sent directly to a robot management server, and stored in the database section tagged for the user of that robot. The best image can then be accessed by the user, rather than multiple, duplicate images. The best image will typically be one where the object fills most of the image, but does not overfill it, and has a higher probability of matching an identified object or hazard than other images of the same object.

Corrective Action in Response to Object Detection.

[0078] FIG. **11** is a flowchart of an embodiment of a method for detecting hazards and taking corrective action. Images are recorded in a buffer memory as the robot moves (**1102**), and are tagged with a timestamp and x, y location coordinates. As described above, the LIDAR can be used to determine the location of the object relative to the robot, and also to determine the robot location on a map. When a jam event occurs, it is recorded with a location (**1104**). A jam event includes anything which adversely affects the operability of the robot, such as clogging, requiring increased brush or movement motor power, trapping of the robot, immobilizing of the robot, etc. Such events can be detected with one or more sensors. The LIDAR can detect that the cleaning robot isn't moving. A current or voltage sensor can detect excessive power being required by the cleaning robot motor for translational movement, or the brush or other cleaning motor. The buffered images corresponding to the jammed location are then transmitted to the remote server (**1106**).

[0079] Corrective action can then be taken (**1108**), such as reversing the direction of rotation of the cleaning brush, allowing the brush to free spin and then backing up the robot, reversing the direction of the robot, increasing the robot brush or translational movement motor power, etc. If the jam event is not corrected, the user is notified (**1110**). The notification can be an indication on the robot app, a separate text message, or any other notification. The user can be directed to take appropriate action, such as clean the brush, remove, empty and replace the dirt container, pick up and move the robot to an open area, etc. The user can optionally be prompted to identify the object at the location of the jam event, and the user identification can be recorded and transmitted to the remote server (**1112**). Thus, the remote server may receive multiple types of tagged images: images tagged as causing a jam that was automatically overcome, images tagged as causing a jam that was not overcome, and images that caused a jam and have been labeled by a user.

[0080] Embodiments provide practical and economic methods and apparatus for asynchronously classifying images provided by a robot. Doing object identification using the processor in a robot in real time would make the robot more expensive. Since the robot takes a fair amount of time to do cleaning, and objects can be bypassed and returned to, real-time decisions are not needed such as would be needed in self-driving cars, for example. In a reconnaissance/exploratory or first cleaning pass, unidentified objects are simply avoided. Images of the object are uploaded over the Internet to a remote object detection and classification system, and the location is indicated by the cleaning robot. When the remote system subsequently returns an object identification or classification, the object can be indicated as something to be avoided, or the cleaning robot can return to the location and clean over the object if it is determined not to be a hazard. The classification of the object need not identify the object, but can simply be an indication that it is a potential hazard to the robot. New images and objects are compared to the tagged hazards database to identify hazards. This eliminates the need for object recognition and classification—the robot will simply know that such an object is a hazard and has impacted other robots adversely.

[0081] In one embodiment, the cleaning robot can obtain additional information about the object. The additional information, or some additional information may be obtained upon the first encounter with the object. Alternately, the cleaning robot may return to the object when object classification is indefinite. The additional information can be additional image views of the object from different directions or angles. Multiple cameras on the cleaning robot can capture different angles, or the cleaning robot can be maneuvered around the object for different views. A bump or pressure sensor can be used with slight contact with the object to determine if it is hard or soft. For example, after detecting initial contact, the robot can continue to move for ½ inch to see if the object compresses or moves. The difference between the object moving (indicating it is hard) and compressing (indicating it is soft) can be determined by the amount of pressure detected on a bump sensor (with, in general, more pressure from a hard, moving object) and/or images or the LIDAR indicating that the object has moved after the robot initiates contact and then withdraws from contact.

Questionnaire

[0082] In one embodiment, a user completes a questionnaire and the answers are used to filter the potential object matches. For example, if the user does not have a pet, dog poop can be eliminated from the possible object classification. Conversely, if a user has a dog, dog poop can be added to the list of potential objects with a higher weighting of likelihood of a match. If a user has kids, toys can be weighted higher, or eliminated if a user doesn't have kids. Indicating birthdays can be used to increase the likelihood weighting of wrapping paper and ribbons around the time of the birthday. Other calendar dates can be used to increase the likelihood weighting, such as wrapping paper or ornaments around Christmas.

[0083] In one embodiment, the type of object detected may change the cleaning mode. For example, the detection of a throw rug on a wood or tile floor can change the brush mode for a vacuum cleaner robot. Different floor types may be stored as images indicating they are not a hazard, and also being tagged with the preferred cleaning mode.

[0084] In one embodiment, the robot may determine the image is too dark, or the remote server may indicate this with a request for a better illuminated image. The robot may have a light source that can be directed to the object and can be turned on. The light source could be visible or IR. Alternately, the robot may communicate via WiFi over a home network with a lighting controller to have a light turned on in the room where the object is located.

Machine Learning

[0085] In one embodiment, machine learning is used to determine image types and whether they are a hazard. A test environment may be set up with multiple known objects. These objects can both be tagged by a human tester, and also can be identified by test robots probing them, running over them, etc. The test objects are selected from a group typically found on the floor of a home, such as socks, wires, papers, dog poop, dog food, string, pencils, etc., etc.

[0086] In one embodiment, the type of room is identified, and the objects are weighted based on their likelihood of being in such a room. For example, a kitchen may be more likely to have food, utensils, etc. A bathroom is more likely to have towels, toothbrushes, etc. A closet is more likely to have socks and other clothing.

CONCLUSION

[0087] While the invention has been described with respect to specific embodiments, one skilled in the art will recognize that numerous modifications are possible. Embodiments of the invention may be realized using a variety of computer systems and communication technologies including but not limited to specific examples described herein.

[0088] Embodiments of the present invention may be realized using any combination of dedicated components and/or programmable processors and/or other programmable devices. The various processes described herein may be implemented on the same processor or different processors in any combination. Where components are described as being configured to perform certain operations, such configuration may be accomplished, e.g., by designing electronic circuits to perform the operation, by programming programmable electronic circuits (such as microprocessors) to perform the operation, or any combination thereof. Further, while the embodiments described above may make reference to specific hardware and software components, those skilled in the art will appreciate that different combinations of hardware and/or software components may also be used and that particular operations described as being implemented in hardware might also be implemented in software or vice versa.

[0089] Computer programs incorporating various features of the present invention may be encoded and stored on various computer readable storage media; suitable media include magnetic disk or tape, optical storage media such as compact disk (CD) or DVD (digital versatile disk), flash memory, and other non-transitory media. Computer readable media encoded with the program code may be packaged with a compatible electronic device, or the program code may be provided separately from electronic devices (e.g., via Internet download or as a separately packaged computer-readable storage medium).

[0090] Thus, although the invention has been described with respect to specific embodiments, it will be appreciated that the invention is intended to cover all modifications and equivalents within the scope of the following claims.

What is claimed is:

1. A mobile cleaning robot comprising:

a robotic apparatus with a housing;

a drive motor mounted in the housing;

a drive system, coupled to the drive motor, for moving the robotic apparatus;

a processor;

a memory;

a distance and object detection sensor;

an image sensor;

a wireless transceiver;

a non-transitory computer readable media, coupled to the processor, containing instructions for:

creating a map of an operating environment using data from the distance and object detection sensor,

performing image processing to recognize objects in images captured by the image sensor,

cropping the objects in the images,

transmitting the cropped images using the wireless transceiver,

tagging a location of the cropped image on the map,

receiving an object classification of the cropped images from a remote object classifier, with the object classification indicating whether the objects are a potential hazard,

returning to and cleaning tagged locations of objects not indicated as a potential hazard.

2. The mobile cleaning robot of claim **1** further comprising:

a cleaning brush mounted in the housing;

a brush motor coupled to the cleaning brush; and

the non-transitory computer readable media further containing instructions for:

turning off the brush motor when the mobile cleaning robot passes over one of the objects.

3. The mobile cleaning robot of claim **1** wherein the turning off the brush motor when the robot passes over one of the objects is performed before the object is classified to indicate whether the object is a potential hazard.

4. The mobile cleaning robot of claim **1** wherein the non-transitory computer readable media further containing instructions for:

slowing down the drive motor when the mobile cleaning robot passes over one of the objects.

5. The mobile cleaning robot of claim **1** wherein indicating whether the object is a potential hazard comprises a confidence rating.

6. The mobile cleaning robot of claim **1** wherein the non-transitory computer readable media further contains instructions for obtaining at least one additional image of the potential object from a different viewpoint.

7. The mobile cleaning robot of claim **1** further comprising:

a pressure sensor for providing a signal corresponding to contact with an object:

wherein the non-transitory computer readable media further comprises instructions for:

maneuvering the cleaning robot to initiate contact by the pressure sensor with the object;

recording the amount of pressure detected by the pressure sensor;

associating data corresponding to the amount of pressure with the object and the location of the object; and

transmitting data corresponding to the amount of pressure.

8. The mobile cleaning robot of claim **1** wherein the non-transitory computer readable media further comprises instructions for:

receiving an impairment indication that the operability of the robot has been impaired;

retrieving pre-impairment images obtained for a period of time prior to the impairment indication; and

transmitting at least a portion of the pre-impairment images for remote storage in a hazards database.

9. A method for operating a mobile cleaning robot comprising:

creating a map of an operating environment using data from a distance and object detection sensor in the mobile cleaning robot;

performing image processing to recognize objects in images captured by an image sensor in the mobile cleaning robot;

cropping the objects in the images;

transmitting the cropped images using a wireless transceiver in the mobile cleaning robot;

tagging locations of the cropped images on the map;

receiving an object classification of the cropped images from a remote object classifier, with the object classification indicating whether the objects are a potential hazard; and

returning to and cleaning tagged locations of objects not indicated as a potential hazard.

10. The method of claim **9** further comprising:

performing one of turning off a brush motor and reducing a mobile cleaning robot speed when the mobile cleaning robot passes over one of the objects.

11. The method of claim **10** wherein the turning off the brush motor when the robot passes over one of the objects is performed before the object is classified to indicate whether the object is a potential hazard.

12. The method of claim **9** further comprising:

maintaining a database of object images with tagged classifications;

comparing the cropped images to the object images in the database; and

obtaining responses from a user questionnaire; and

limiting the comparing to object images in accordance with the responses from the user questionnaire.

13. The method of claim **9** wherein the object classification indicates a type of object.

14. The method of claim **9** further comprising:

providing at least one cropped image to a user device;

receiving from the user device an user hazard indication of whether the at least one cropped image is a hazard; and

storing the user hazard indication in association with the cropped image in a hazard database; and

comparing subsequent images to the stored cropped image in the hazard database; and

providing a hazard indication probability for each of the subsequent images based at least in part on the degree of similarity to the stored cropped image in the hazard database.

15. The method claim **9** further comprising:

receiving an impairment indication that the operability of the robot has been impaired;

retrieving pre-impairment images obtained for a period of time prior to the impairment indication; and

transmitting at least a portion of the pre-impairment images for remote storage in a hazards database.

16. The method of claim **15** further comprising:

controlling the cleaning robot to take a remedial action to attempt to overcome the impairment;

if the remedial action is successful, tagging the pre-impairment images with a first hazard probability; and

if the remedial action is unsuccessful, tagging the pre-impairment image with a second hazard probability, wherein the second hazard probability is higher than the first hazard probability.

17. The method of claim **9** further comprising:

returning to and cleaning tagged locations of objects indicated as a potential hazard with a hazard probability below a set level; and

receiving a user input to adjust the set level.

18. The method of claim **9** further comprising:

maneuvering the cleaning robot to initiate contact by a pressure sensor with an object;

recording the amount of pressure detected by the pressure sensor;

associating data corresponding to the amount of pressure with an image of the object and the location of the object; and

transmitting data corresponding to the amount of pressure to a remote database.

19. The method of claim **9** further comprising:

maneuvering the cleaning robot to initiate contact with an object;

recording whether the object moved as a result of the contact;

associating data corresponding to the amount of movement of the object with an image of the object and the location of the object; and

transmitting data corresponding to the amount of movement to a remote database.

20. A mobile cleaning robot comprising:

a housing;

a drive motor mounted in the housing;

a drive system, coupled to the drive motor, for moving the mobile cleaning robot;

a cleaning element, mounted in the housing;

a processor;

a distance and object detection sensor comprising a source providing collimated light output in an emitted light beam and a detector sensor operative to detect a reflected light beam from the emitted light beam incident on an object, and further comprising:

a rotating mount to which said source and said detector sensor are attached;

an angular orientation sensor operative to detect an angular orientation of the rotating mount;

a first non-transitory, computer readable media including instructions for

computing distance between the rotating mount and the object,

determining a direction of the stationary object relative to the robotic device using the angular orientation of the rotating mount, and applying a simultaneous localization and mapping (SLAM) algorithm to the distance and the direction to determine a location of the robotic device and to map an operating environment;

a second non-transitory computer readable media, coupled to the processor, containing instructions for:

creating a map of an operating environment using data from the distance and object detection sensor,

performing image processing to recognize objects in images captured by the image sensor,

cropping the objects in the images,

transmitting the cropped images using the wireless transceiver,

tagging a location of the cropped image on the map, receiving an object classification of the cropped images from a remote object classifier, with the object classification indicating whether the objects are a potential hazard,

returning to and cleaning tagged locations of objects not indicated as a potential hazard.

* * * * *