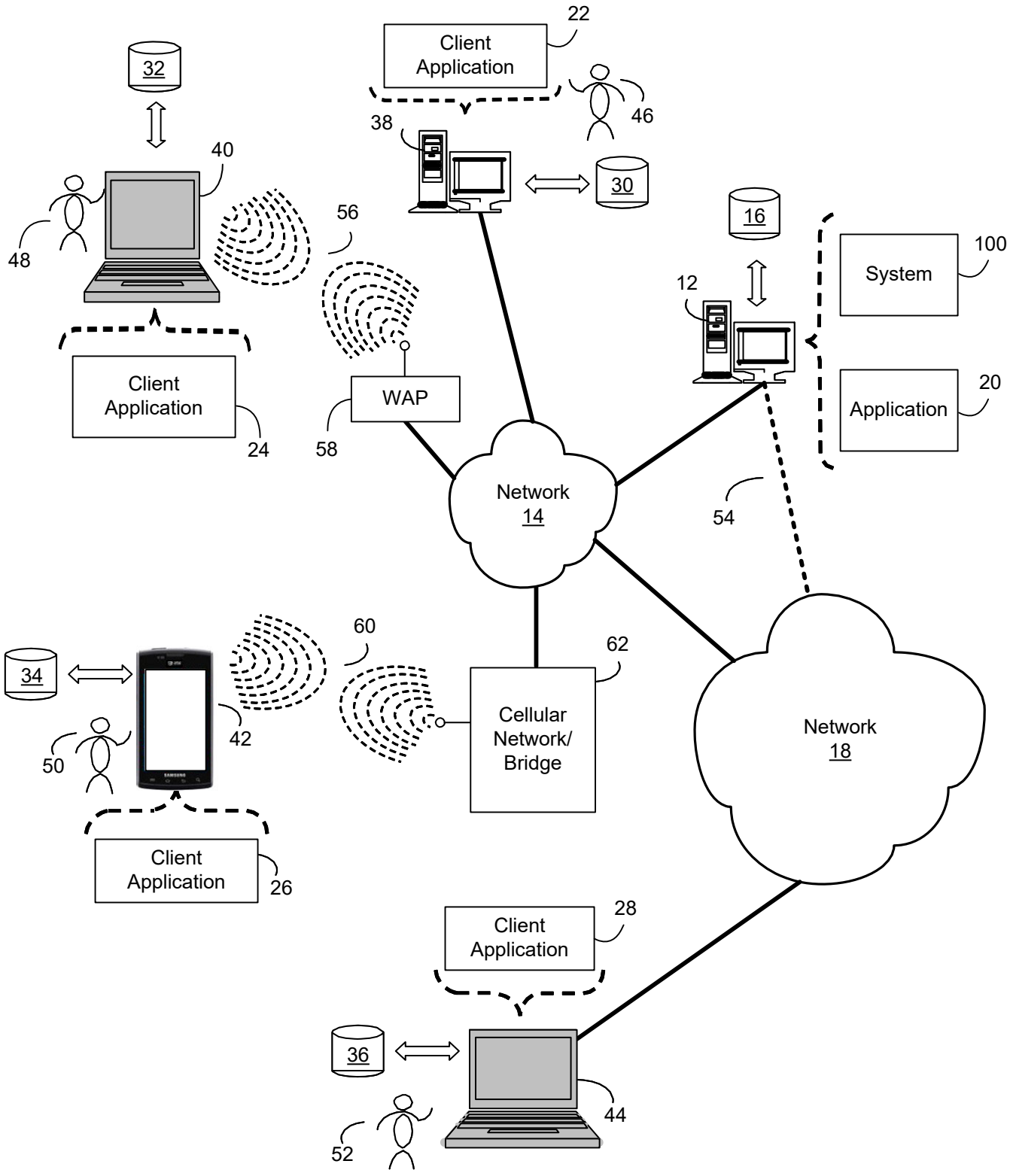

FIG. 1

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant

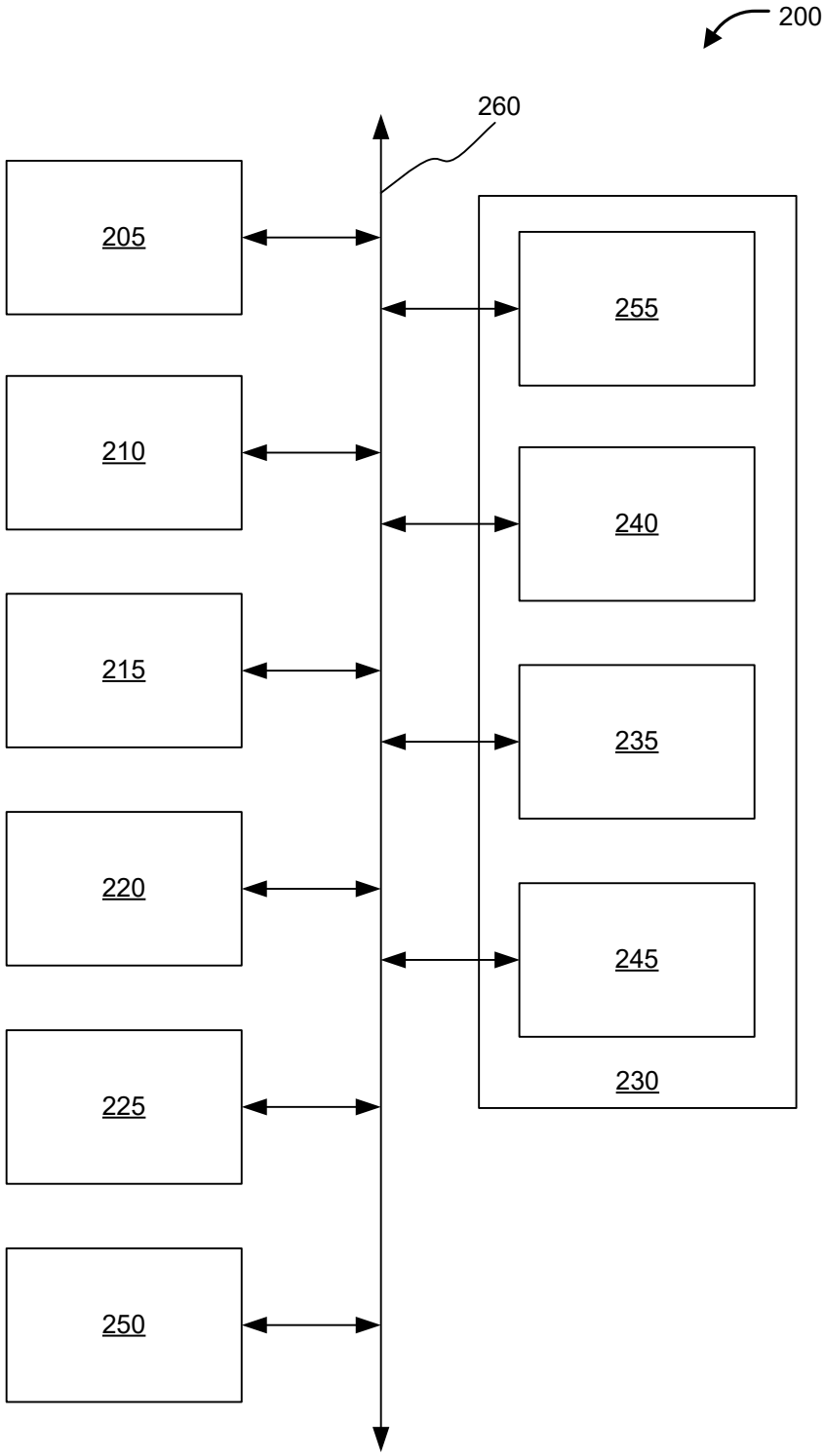

FIG. 2

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant

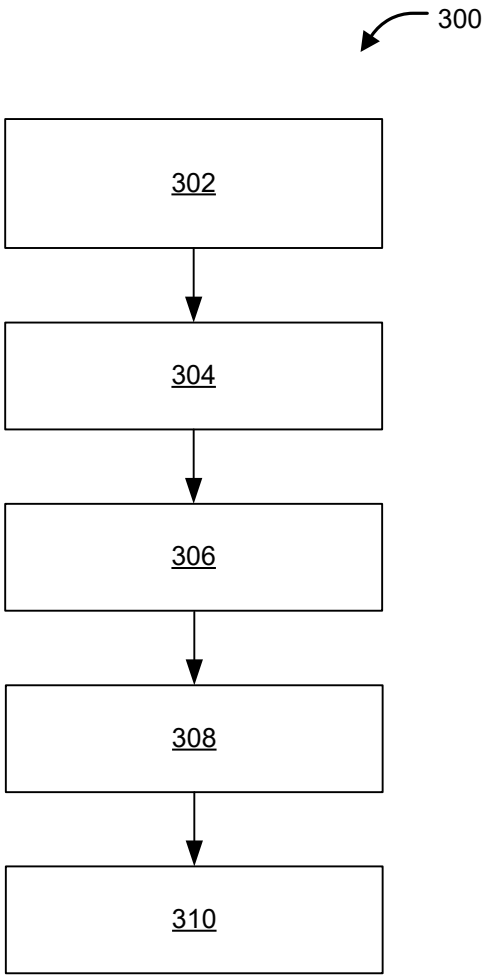

FIG. 3

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant

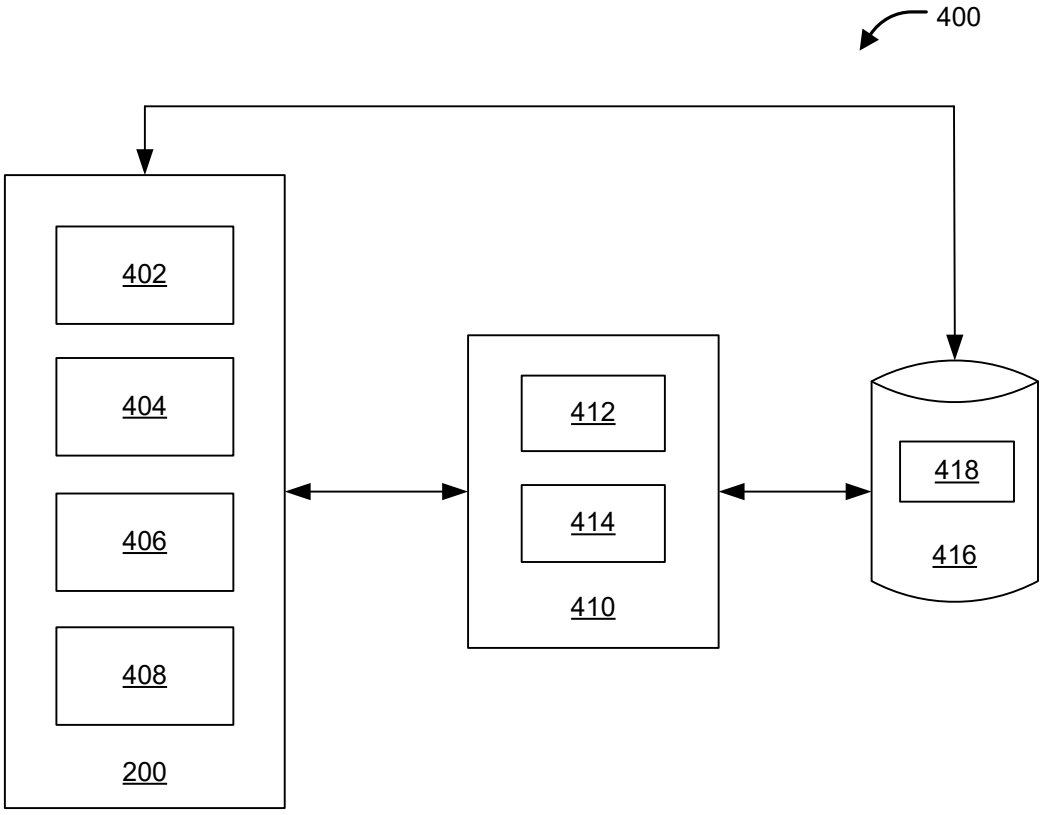

FIG. 4

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant

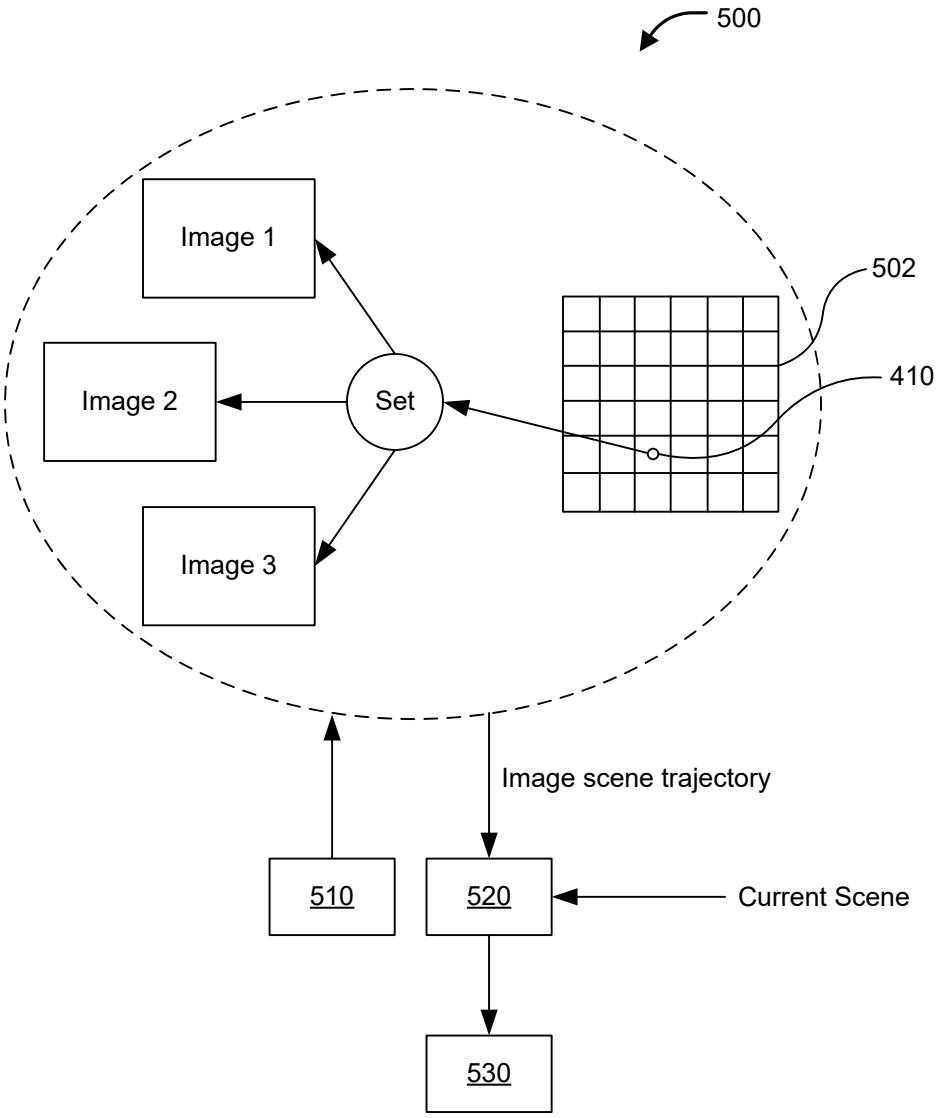

FIG. 5

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant

FORM 2
THE PATENTS ACT, 1970
(39 of 1970)
&
The Patents Rules, 2003
COMPLETE SPECIFICATION
[See section 10 and rule 13]

TITLE OF THE INVENTION: **SYSTEM AND METHOD FOR DETERMINING ERROR IN LOCALIZATION IN NAVIGATION OF MOBILE ROBOT**

1. APPLICANT

(a) NAME - **ADDVERB TECHNOLOGIES PRIVATE LIMITED**

(b) NATIONALITY - **Indian**

(c) ADDRESS - **Plot No. 5, Sector-156, Noida-201301**

PREAMBLE TO THE DESCRIPTION

The following specification particularly describes the invention and the manner in which it is to be performed.

SYSTEM AND METHOD FOR DETERMINING ERROR IN LOCALIZATION IN NAVIGATION OF MOBILE ROBOT

FIELD OF THE PRESENT DISCLOSURE

[0001] The present disclosure generally relates to autonomous guided vehicles, such as a mobile robot, implemented to move in an environment, and particularly to a system and method for determining an error in localization in navigation of a mobile robot in an environment by determining localization error in the navigation thereof due to slow moving objects in the environment and/or change in discrete features in the environment.

BACKGROUND

[0002] Autonomous guided vehicles (AGVs), also known as mobile robots, are increasingly being employed for transporting goods and materials from one place to another in constrained environments, such as a factory or a warehouse. For example, mobile robots are used in warehouse environments to assist with inventory management by transporting goods from one area of the warehouse to another. In the warehouse, the mobile robot may travel from a loading area to a dropping area based on a control system and without intervention from users. In a manufacturing plant, the mobile robots can transport items such as heavy vehicle components like engines, chassis, etc. along a route on a floor of the manufacturing plant to deliver the payload from one location to another or to allow various manufacturing operations to be performed thereon. Mobile robots may offer the ability to carry payloads too heavy for a person to carry and without the supervision of a person, while also offering the flexibility to be reconfigured to follow a different route or carry different types of payloads.

[0003] Although localization and navigation of mobile robots in a defined environment is a robust technology; changes in the environment post the mapping stage and presence of slow moving objects result in erroneous localization. In particular, during mapping stage, a simultaneous, fully connected mapping stack (unordered map) is developed which corresponds to the scenic view vis-a-vis the robot pose. Once the coordinate map and the scenic

map are simultaneously created and stored in a hash table; the mobile robot may implement the same and tries to repeatedly match a current scenic view it sees with the scenic view it wants to see (from the stored hash table). It may be understood that, herein, the scenic views are accessed from the stored hash table, based on the coordinates of the mobile robot. Now due to the change in discrete features in the environment and/or the presence of slow moving objects in the environment, the mobile robot may not be able to find a decent match for the current scenic view and thus may not be able to localize itself in the environment.

[0004] The problem of re-localizing the robot once lost is a known problem where significant changes in the environment are detected and the robot autonomously tries to re-localize itself within the environment. However, the localization error due to slow moving objects in the environment or the change in discrete features in the environment leading to drift based errors in robot localization has not been particularly solved. Such errors in localization might lead to catastrophic accidents as the mobile robot may hit other objects. Therefore, it is desirable to describe a need to remap at intermediate time instances such that the error in subsequent localization and mapping is mitigated.

[0005] Therefore, in light of the foregoing discussion, there exists a need to overcome problems associated with conventional techniques and provide systems and/or methods for determining an error in localization in navigation of mobile robots, specifically to raise requisite flags in case of localization error due to slow moving objects in the environment or the change in discrete features in the environment, suggesting a need of remapping, and thereby ensuring less navigation errors for the mobile robots.

SUMMARY

[0006] In an aspect of the present disclosure, a method for determining an error in localization in navigation of a mobile robot in an environment is provided. The method comprises implementing a mapping stack comprising one or more scenic views corresponding to respective one or more poses for the mobile robot, for each of mapped coordinate in the environment. The method further comprises capturing, by the mobile robot, a current scenic view with information about a current pose and a current coordinate thereof, in the environment. The method further

comprises comparing the current scenic view with the scenic view from the mapping stack having the corresponding pose and the corresponding mapped coordinate matching the current pose and the current coordinate. The method further comprises calculating a matching score for the current scenic view based on the comparison. The method further comprises generating a localization error flag in case of the matching score being below a predefined threshold.

[0007] In one or more embodiments, the method further comprises configuring the mobile robot to perform one or more secondary manoeuvres, in case of generation of the localization error flag; and confirming a localization error in the navigation of the mobile robot based on the one or more secondary manoeuvres.

[0008] In one or more embodiments, the method further comprises appending the current scenic view with information about the current pose and the current coordinate thereof in the mapping stack, based on confirmation of the localization error in the navigation of the mobile robot.

[0009] In one or more embodiments, the method further comprises capturing, by the mobile robot, a series of current scenic views with information about the corresponding current poses and the corresponding current coordinates, over a trajectory of the mobile robot in the environment; comparing each of the series of current scenic views with the correlative scenic view from the mapping stack, having the corresponding pose and the corresponding mapped coordinate matching the respective current pose and the respective current coordinate; determining an individual matching score for each of the series of current scenic views based on each of the comparison; and calculating the matching score as a discounted sum of the individual matching scores.

[0010] In one or more embodiments, calculating the matching score as the discounted sum of the individual matching scores is based on: a weighted average of the individual matching scores, with a later of the current scenic view of the series of current scenic views being given a higher weight in comparison to former of the current scenic view of the series of current scenic views; and a rate of reduction of the individual matching scores from the former of the current scenic view of the series of current scenic views to the later of the current scenic view of the series of current scenic views being substantially constant in relation to the series of current scenic views.

[0011] In another aspect of the present disclosure, a system for determining an error

in localization in navigation of a mobile robot in an environment is provided. The system comprises a database having stored a mapping stack comprising one or more scenic views corresponding to respective one or more poses for the mobile robot, for each of mapped coordinate in the environment. The system further comprises a processing arrangement in communication with the database to access the mapping stack. The processing arrangement is configured to instruct the mobile robot to capture a current scenic view with information about a current pose and a current coordinate thereof, in the environment. The processing arrangement is further configured to compare the current scenic view with the scenic view from the mapping stack having the corresponding pose and the corresponding mapped coordinate matching the current pose and the current coordinate. The processing arrangement is further configured to calculate a matching score for the current scenic view based on the comparison. The processing arrangement is further configured to generate a localization error flag in case of the matching score being below a predefined threshold.

[0012] In one or more embodiments, the processing arrangement is further configured to: instruct the mobile robot to perform one or more secondary manoeuvres, in case of generation of the localization error flag; and confirm a localization error in the navigation of the mobile robot based on the one or more secondary manoeuvres.

[0013] In one or more embodiments, the processing arrangement is further configured to append the current scenic view with information about the current pose and the current coordinate thereof in the mapping stack stored in the database, based on confirmation of the localization error in the navigation of the mobile robot.

[0014] In one or more embodiments, the processing arrangement is further configured to: instruct the mobile robot to capture a series of current scenic views with information about the corresponding current poses and the corresponding current coordinates, over a trajectory of the mobile robot in the environment; compare each of the series of current scenic views with the correlative scenic view from the mapping stack, having the corresponding pose and the corresponding mapped coordinate matching the respective current pose and the respective current coordinate; determine an individual matching score for each of the series of current scenic views based on each of the comparison; and calculate the matching score as a discounted sum of the individual matching scores.

[0015] In one or more embodiments, calculating the matching score as the discounted sum of the individual matching scores is based on: a weighted average of the individual matching scores, with a later of the current scenic view of the series of current scenic views being given a higher weight in comparison to former of the current scenic view of the series of current scenic views; and a rate of reduction of the individual matching scores from the former of the current scenic view of the series of current scenic views to the later of the current scenic view of the series of current scenic views being substantially constant in relation to the series of current scenic views.

[0016] The foregoing summary is illustrative only and is not intended to be in any way limiting. In addition to the illustrative aspects, embodiments, and features described above, further aspects, embodiments, and features will become apparent by reference to the drawings and the following detailed description.

BRIEF DESCRIPTION OF THE FIGURES

[0017] For a more complete understanding of example embodiments of the present disclosure, reference is now made to the following descriptions taken in connection with the accompanying drawings in which:

[0018] FIG. 1 illustrates a schematic of an exemplary computing system that may reside on and may be executed by a computer, and which may be connected to a network, in accordance with one or more embodiments of the present disclosure;

[0019] FIG. 2 illustrates a schematic of an exemplary processing arrangement, in accordance with one or more embodiments of the present disclosure;

[0020] FIG. 3 illustrates a flowchart listing steps involved in a method for determining an error in localization in navigation of a mobile robot in an environment, in accordance with one or more embodiments of the present disclosure;

[0021] FIG. 4 illustrates a schematic of a system for determining an error in localization in navigation of a mobile robot in an environment, in accordance with one or more embodiments of the present disclosure; and

[0022] FIG. 5 illustrates a process flow diagram for determination of localization

error in the navigation of the mobile robot in the environment, in accordance with one or more embodiments of the present disclosure.

DETAILED DESCRIPTION

[0023] In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present disclosure. It will be apparent, however, to one skilled in the art that the present disclosure is not limited to these specific details.

[0024] Reference in this specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present disclosure. The appearance of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment, nor are separate or alternative embodiments mutually exclusive of other embodiments. Further, the terms "a" and "an" herein do not denote a limitation of quantity, but rather denote the presence of at least one of the referenced items. Moreover, various features are described which may be exhibited by some embodiments and not by others. Similarly, various requirements are described which may be requirements for some embodiments but not for other embodiments.

[0025] Furthermore, in the following detailed description of the present disclosure, numerous specific details are set forth in order to provide a thorough understanding of the present disclosure. However, it will be understood that the present disclosure may be practiced without these specific details. In other instances, well-known methods, procedures, components, and circuits have not been described in detail so as not to unnecessarily obscure aspects of the present disclosure.

[0026] Embodiments described herein may be discussed in the general context of computer-executable instructions residing on some form of computer-readable storage medium, such as program modules, executed by one or more computers or other devices. By way of example, and not limitation, computer-readable storage media may comprise non-transitory computer-readable storage media and communication media; non-transitory computer-readable

media include all computer-readable media except for a transitory, propagating signal. Generally, program modules include routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types. The functionality of the program modules may be combined or distributed as desired in various embodiments.

[0027] Some portions of the detailed description that follows are presented and discussed in terms of a process or method. Although steps and sequencing thereof are disclosed in figures herein describing the operations of this method, such steps and sequencing are exemplary. Embodiments are well suited to performing various other steps or variations of the steps recited in the flowchart of the figure herein, and in a sequence other than that depicted and described herein. Some portions of the detailed descriptions that follow are presented in terms of procedures, logic blocks, processing, and other symbolic representations of operations on data bits within a computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. In the present disclosure, a procedure, logic block, process, or the like, is conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those utilizing physical manipulations of physical quantities. Usually, although not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as transactions, bits, values, elements, symbols, characters, samples, pixels, or the like.

[0028] In some implementations, any suitable computer usable or computer readable medium (or media) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. The computer-usable, or computer-readable, storage medium (including a storage device associated with a computing device) may be, for example, but is not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer-readable medium may include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fibre, a portable compact disc read-only memory (CD-ROM), an optical storage device, a digital versatile disk (DVD), a static random

access memory (SRAM), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, a media such as those supporting the internet or an intranet, or a magnetic storage device. Note that the computer-usable or computer-readable medium could even be a suitable medium upon which the program is stored, scanned, compiled, interpreted, or otherwise processed in a suitable manner, if necessary, and then stored in a computer memory. In the context of the present disclosure, a computer-usable or computer-readable, storage medium may be any tangible medium that can contain or store a program for use by or in connection with the instruction execution system, apparatus, or device.

[0029] In some implementations, a computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. In some implementations, such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. In some implementations, the computer readable program code may be transmitted using any appropriate medium, including but not limited to the internet, wireline, optical fibre cable, RF, etc. In some implementations, a computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device.

[0030] In some implementations, computer program code for carrying out operations of the present disclosure may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, or either source code or object code written in any combination of one or more programming languages, including an object-oriented programming language such as Java®, Smalltalk, C++ or the like. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. However, the computer program code for carrying out operations of the present disclosure may also be written in conventional procedural programming languages, such as the "C" programming language, PASCAL, or similar programming languages, as well as in scripting languages such as JavaScript, PERL, or Python. In present implementations, the used language for training may be one of Python, Tensorflow™, Bazel, C, C++. Further, decoder in user device (as will be discussed) may use C, C++ or any processor specific ISA. Furthermore, assembly code inside C/C++ may be utilized for specific

operation. Also, ASR (automatic speech recognition) and G2P decoder along with entire user system can be run in embedded Linux (any distribution), Android, iOS, Windows, or the like, without any limitations. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the internet using an Internet Service Provider). In some implementations, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGAs) or other hardware accelerators, micro-controller units (MCUs), or programmable logic arrays (PLAs) may execute the computer readable program instructions/code by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present disclosure.

[0031] In some implementations, the flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of apparatus (systems), methods and computer program products according to various implementations of the present disclosure. Each block in the flowchart and/or block diagrams, and combinations of blocks in the flowchart and/or block diagrams, may represent a module, segment, or portion of code, which comprises one or more executable computer program instructions for implementing the specified logical function(s)/act(s). These computer program instructions may be provided to a processor of a general-purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the computer program instructions, which may execute via the processor of the computer or other programmable data processing apparatus, create the ability to implement one or more of the functions/acts specified in the flowchart and/or block diagram block or blocks or combinations thereof. It should be noted that, in some implementations, the functions noted in the block(s) may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved.

[0032] In some implementations, these computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data

processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the function/act specified in the flowchart and/or block diagram block or blocks or combinations thereof.

[0033] In some implementations, the computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed (not necessarily in a particular order) on the computer or other programmable apparatus to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions/acts (not necessarily in a particular order) specified in the flowchart and/or block diagram block or blocks or combinations thereof.

[0034] Referring now to the example implementation of FIG. 1, illustrated is a computing system 100 that may reside on and may be executed by a computer (e.g., computer 12), which may be connected to a network (e.g., network 14) (e.g., the internet or a local area network). Examples of computer 12 may include, but are not limited to, a personal computer(s), a laptop computer(s), mobile computing device(s), a server computer, a series of server computers, a mainframe computer(s), or a computing cloud(s). In some implementations, each of the aforementioned may be generally described as a computing device. In certain implementations, a computing device may be a physical or virtual device. In many implementations, a computing device may be any device capable of performing operations, such as a dedicated processor, a portion of a processor, a virtual processor, a portion of a virtual processor, a portion of a virtual device, or a virtual device. In some implementations, a processor may be a physical processor or a virtual processor. In some implementations, a virtual processor may correspond to one or more parts of one or more physical processors. In some implementations, the instructions/logic may be distributed and executed across one or more processors, virtual or physical, to execute the instructions/logic. Computer 12 may execute an operating system, for example, but not limited to, Microsoft® Windows®; Mac® OS X®; Red Hat® Linux®, or a custom operating system. (Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries or both; Mac and OS X are registered trademarks of Apple Inc. in the United States, other countries or both; Red Hat is a registered trademark of Red Hat Corporation in the United States, other countries or both; and Linux is a registered trademark of Linus Torvalds in the United

States, other countries or both).

[0035] In some implementations, the instruction sets and subroutines of computing system 100, which may be stored on storage device, such as storage device 16, coupled to computer 12, may be executed by one or more processors (not shown) and one or more memory architectures included within computer 12. In some implementations, storage device 16 may include but is not limited to: a hard disk drive; a flash drive, a tape drive; an optical drive; a RAID array (or other array); a random-access memory (RAM); and a read-only memory (ROM). In some implementations, network 14 may be connected to one or more secondary networks (e.g., network 18), examples of which may include but are not limited to: a local area network; a wide area network; or an intranet, for example.

[0036] In some implementations, computer 12 may include a data store, such as a database (e.g., relational database, object-oriented database, triplestore database, etc.) and may be located within any suitable memory location, such as storage device 16 coupled to computer 12. In some implementations, data, metadata, information, etc. described throughout the present disclosure may be stored in the data store. In some implementations, computer 12 may utilize any known database management system such as, but not limited to, DB2, in order to provide multi-user access to one or more databases, such as the above noted relational database. In some implementations, the data store may also be a custom database, such as, for example, a flat file database or an XML database. In some implementations, any other form(s) of a data storage structure and/or organization may also be used. In some implementations, computing system 100 may be a component of the data store, a standalone application that interfaces with the above noted data store and/or an applet / application that is accessed via client applications 22, 24, 26, 28. In some implementations, the above noted data store may be, in whole or in part, distributed in a cloud computing topology. In this way, computer 12 and storage device 16 may refer to multiple devices, which may also be distributed throughout the network.

[0037] In some implementations, computer 12 may execute application 20 for navigation of a mobile robot in an environment. In some implementations, computing system 100 and/or application 20 may be accessed via one or more of client applications 22, 24, 26, 28. In some implementations, computing system 100 may be a standalone application, or may be an applet / application / script / extension that may interact with and/or be executed within application

20, a component of application 20, and/or one or more of client applications 22, 24, 26, 28. In some implementations, application 20 may be a standalone application, or may be an applet / application / script / extension that may interact with and/or be executed within computing system 100, a component of computing system 100, and/or one or more of client applications 22, 24, 26, 28. In some implementations, one or more of client applications 22, 24, 26, 28 may be a standalone application, or may be an applet / application / script / extension that may interact with and/or be executed within and/or be a component of computing system 100 and/or application 20. Examples of client applications 22, 24, 26, 28 may include, but are not limited to, a standard and/or mobile web browser, an email application (e.g., an email client application), a textual and/or a graphical user interface, a customized web browser, a plugin, an Application Programming Interface (API), or a custom application. The instruction sets and subroutines of client applications 22, 24, 26, 28, which may be stored on storage devices 30, 32, 34, 36, coupled to user devices 38, 40, 42, 44, may be executed by one or more processors and one or more memory architectures incorporated into user devices 38, 40, 42, 44.

[0038] In some implementations, one or more of storage devices 30, 32, 34, 36, may include but are not limited to: hard disk drives; flash drives, tape drives; optical drives; RAID arrays; random access memories (RAM); and read-only memories (ROM). Examples of user devices 38, 40, 42, 44 (and/or computer 12) may include, but are not limited to, a personal computer (e.g., user device 38), a laptop computer (e.g., user device 40), a smart/data-enabled, cellular phone (e.g., user device 42), a notebook computer (e.g., user device 44), a tablet (not shown), a server (not shown), a television (not shown), a smart television (not shown), a media (e.g., video, photo, etc.) capturing device (not shown), and a dedicated network device (not shown). User devices 38, 40, 42, 44 may each execute an operating system, examples of which may include but are not limited to, Android®, Apple® iOS®, Mac® OS X®; Red Hat® Linux®, or a custom operating system.

[0039] In some implementations, one or more of client applications 22, 24, 26, 28 may be configured to effectuate some or all of the functionality of computing system 100 (and vice versa). Accordingly, in some implementations, computing system 100 may be a purely server-side application, a purely client-side application, or a hybrid server-side / client-side application that is cooperatively executed by one or more of client applications 22, 24, 26, 28 and/or computing system 100.

[0040] In some implementations, one or more of client applications 22, 24, 26, 28 may be configured to effectuate some or all of the functionality of application 20 (and vice versa). Accordingly, in some implementations, application 20 may be a purely server-side application, a purely client-side application, or a hybrid server-side / client-side application that is cooperatively executed by one or more of client applications 22, 24, 26, 28 and/or application 20. As one or more of client applications 22, 24, 26, 28, computing system 100, and application 20, taken singly or in any combination, may effectuate some or all of the same functionality, any description of effectuating such functionality via one or more of client applications 22, 24, 26, 28, computing system 100, application 20, or combination thereof, and any described interaction(s) between one or more of client applications 22, 24, 26, 28, computing system 100, application 20, or combination thereof to effectuate such functionality, should be taken as an example only and not to limit the scope of the disclosure.

[0041] In some implementations, one or more of users 46, 48, 50, 52 may access computer 12 and computing system 100 (e.g., using one or more of user devices 38, 40, 42, 44) directly through network 14 or through secondary network 18. Further, computer 12 may be connected to network 14 through secondary network 18, as illustrated with phantom link line 54. Computing system 100 may include one or more user interfaces, such as browsers and textual or graphical user interfaces, through which users 46, 48, 50, 52 may access computing system 100.

[0042] In some implementations, the various user devices may be directly or indirectly coupled to network 14 (or network 18). For example, user device 38 is shown directly coupled to network 14 via a hardwired network connection. Further, user device 44 is shown directly coupled to network 18 via a hardwired network connection. User device 40 is shown wirelessly coupled to network 14 via wireless communication channel 56 established between user device 40 and wireless access point (i.e., WAP) 58, which is shown directly coupled to network 14. WAP 58 may be, for example, an IEEE 802.11a, 802.11b, 802.11g, Wi-Fi®, RFID, and/or Bluetooth™ (including Bluetooth™ Low Energy) device that is capable of establishing wireless communication channel 56 between user device 40 and WAP 58. User device 42 is shown wirelessly coupled to network 14 via wireless communication channel 60 established between user device 42 and cellular network / bridge 62, which is shown directly coupled to network 14.

[0043] In some implementations, some or all of the IEEE 802.11x specifications may

use Ethernet protocol and carrier sense multiple access with collision avoidance (i.e., CSMA/CA) for path sharing. The various 802.11x specifications may use phase-shift keying (i.e., PSK) modulation or complementary code keying (i.e., CCK) modulation, for example, Bluetooth™ (including Bluetooth™ Low Energy) is a telecommunications industry specification that allows, e.g., mobile phones, computers, smart phones, and other electronic devices to be interconnected using a short-range wireless connection. Other forms of interconnection (e.g., Near Field Communication (NFC)) may also be used.

[0044] For the purposes of the present disclosure, the computing system 100 may include a processing arrangement. Herein, FIG. 2 is a block diagram of an example of a processing arrangement 200 capable of implementing embodiments according to the present disclosure. The processing arrangement 200 is implemented for issuing commands for managing and controlling operations of a mobile robot; and in particular for determination of localization error for supporting navigation of a mobile robot in an environment (as will be described later in more detail). Herein, the environment may be a warehouse environment, a manufacturing plant and the like; in which the mobile robots are typically implemented. In one embodiment, the application 20 for aiding the mobile robot to navigate the environment as described above may be executed as a part of the processing arrangement 200 as described herein. Thereby, for example in case of a warehouse, the computing system 100 may be a broader system such as the warehouse management system (WMS) as known in the art, in which the processing arrangement 200 may be executed for aiding a mobile robot to navigate an environment. Hereinafter, the terms “computing system 100” and “processing arrangement 200” have been broadly interchangeably used to represent means for aiding a mobile robot to navigate an environment, without any limitations.

[0045] In one or more embodiments, as illustrated in FIG. 2, the processing arrangement 200 includes a processing unit 205 for running software applications (such as, the application 20 of FIG. 1) and optionally an operating system. Memory 210 stores applications and data for use by the processing unit 205. Storage 215 provides non-volatile storage for applications and data and may include fixed disk drives, removable disk drives, flash memory devices, and CD-ROM, DVD-ROM or other optical storage devices. An optional user input device 220 includes devices that communicate user inputs from one or more users to the processing arrangement 200 and may include keyboards, mice, joysticks, touch screens, etc. A communication or network interface 225 is provided which allows the processing arrangement 200 to communicate with other

computer systems via an electronic communications network, including wired and/or wireless communication and including an Intranet or the Internet. In one embodiment, the processing arrangement 200 receives instructions and user inputs from a remote computer through communication interface 225. Communication interface 225 can comprise a transmitter and receiver for communicating with remote devices. An optional display device 250 may be provided which can be any device capable of displaying visual information in response to a signal from the processing arrangement 200. The components of the processing arrangement 200, including the processing unit 205, the memory 210, the data storage 215, the user input devices 220, the communication interface 225, and the display device 250, may be coupled via one or more data buses 260.

[0046] As illustrated in FIG. 2, a graphics system 230 may be coupled with the data bus 260 and the components of the processing arrangement 200. The graphics system 230 may include a physical graphics processing unit (GPU) 235 and graphics memory. The GPU 235 generates pixel data for output images from rendering commands. The physical GPU 235 can be configured as multiple virtual GPUs that may be used in parallel (concurrently) by a number of applications or processes executing in parallel. For example, mass scaling processes for rigid bodies or a variety of constraint solving processes may be run in parallel on the multiple virtual GPUs. Graphics memory may include a display memory 240 (e.g., a framebuffer) used for storing pixel data for each pixel of an output image. In another embodiment, the display memory 240 and/or additional memory 245 may be part of the memory 210 and may be shared with the processing unit 205. Alternatively, the display memory 240 and/or additional memory 245 can be one or more separate memories provided for the exclusive use of the graphics system 230. In another embodiment, graphics system 230 includes one or more additional physical GPUs 255, similar to the GPU 235. Each additional GPU 255 may be adapted to operate in parallel with the GPU 235. Each additional GPU 255 generates pixel data for output images from rendering commands. Each additional physical GPU 255 can be configured as multiple virtual GPUs that may be used in parallel (concurrently) by a number of applications or processes executing in parallel, e.g., processes that solve constraints. Each additional GPU 255 can operate in conjunction with the GPU 235, for example, to simultaneously generate pixel data for different portions of an output image, or to simultaneously generate pixel data for different output images. Each additional GPU 255 can be located on the same circuit board as the GPU 235, sharing a connection with the

GPU 235 to the data bus 260, or each additional GPU 255 can be located on another circuit board separately coupled with the data bus 260. Each additional GPU 255 can also be integrated into the same module or chip package as the GPU 235. Each additional GPU 255 can have additional memory, similar to the display memory 240 and additional memory 245, or can share the memories 240 and 245 with the GPU 235. It is to be understood that the circuits and/or functionality of GPU as described herein could also be implemented in other types of processors, such as general-purpose or other special-purpose coprocessors, or within a CPU.

[0047] Referring to FIG. 3, illustrated is a flowchart listing steps involved in a method 300 for determining an error in localization in navigation of a mobile robot in an environment. The steps of the method 300 are implemented by a system, as illustrated in FIG. 4. In particular, FIG. 4 illustrates a schematic of a system 400 for determining an error in localization in navigation of a mobile robot 410 in an environment, in accordance with one or more embodiments of the present disclosure. Herein, the system 400 is specifically implemented for determining of localization error in the navigation of the mobile robot 410 in the environment due to slow moving objects in the environment and/or the change in discrete features in the environment, which, in turn, may then be utilized to assess a need of remapping of the given environment. In one or more embodiments, the system 400 implements the processing arrangement 200 as described in the preceding paragraphs for the said purpose. The mobile robot 410 may be utilized for various operations in the environment, like transferring of goods, such as cartons, in the work area, which is typical, e.g., for the warehouse environment. Further, it may be appreciated that the environment (not depicted) may be an entire work area or part of the work area, e.g., in a warehouse environment or the like.

[0048] The embodiments of the present disclosure have been described with reference to the mobile robot 410 in terms of the problem being solved, and with reference to the mobile robot 410 as part of the disclosed solution. In particular, the mobile robot 410 may be controlled by the processing arrangement 200 by using a fleet management system (FMS) as known in the art and thus not described herein for the brevity of the present disclosure. For purposes of the present disclosure, the mobile robot 410 may be defined as a robot having the capability to move around in the environment and is not fixed to one physical location, and which is autonomous and thus capable of navigating the environment without the need for external physical or electro-mechanical guidance devices.

[0049] In embodiments of the present disclosure, the mobile robot 410 may include a drive arrangement (generally represented by reference numeral 412) which may provide omni-directional and/or holonomic motion control of the mobile robot 410. As used herein the term "omni-directional" refers to the ability to move in substantially any planar direction, i.e., side-to-side (lateral), forward/back, and rotational. Furthermore, the term "holonomic" is used in a manner substantially consistent with the literature use of the term and refers to the ability to move in a planar direction with three planar degrees of freedom, i.e., two translations and one rotation. Hence, the mobile robot 410 may have the ability to move in a planar direction at a velocity made up of substantially any proportion of the three planar velocities (forward/back, lateral, and rotational), as well as the ability to change these proportions in a substantially continuous manner. Accordingly, the drive arrangement 412 may allow for changing a pose of the mobile robot 410, as would be required for navigation of the mobile robot 410 in the environment; for example, for following a defined trajectory thereby. Such drive arrangement 412 may be contemplated by a person skilled in the art of robotics and this not discussed further herein.

[0050] The mobile robot 410 may also include an image capturing device 414 which may be configured to capture images (scenic views/ scenes) of the environment when triggered (e.g., by the processing arrangement 200). In the present examples, the image capturing device 414 may be in the form of a digital camera, an infrared sensor, a LIDAR (LIght Detection And Ranging) sensor, and the like including combinations thereof, without any limitation. In the present examples, the image capturing device 414 has been described in the form of a CMOS sensor, like the camera as known in the art. The image capturing device 414 in the form of a camera, which is known for its low price, ease of use and ability to capture in abundance information, is suitable for vision-based mobile robot navigation as per the embodiments of the present disclosure. The image capturing device 414 may have a certain field-of-view (FOV) and may accordingly capture images of sections of the environment. In some examples, the processing arrangement 200 may be configured to implement some known image processing techniques to correct artifacts in the captured images, warping of the captured images, and the like, as may be contemplated by a person skilled in the art.

[0051] As illustrated in FIG. 4, the system 400 includes a database 416. The database 416 includes a mapping stack (represented by reference numeral 418) stored therein. Herein, the mapping stack 418 may include one or more scenic views corresponding to respective one or more

poses for the mobile robot 410, for each of mapped coordinate in the environment. The mapping stack 418 may generally be in the form of an unorganized data structure which provides a simultaneous fully connected graph corresponding to the scenic view vis-a-vis a pose of the mobile robot 410. As used herein, the term "database" may refer to an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just a database. Data within the most common types of databases in operation today is typically modelled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. In embodiments of the present disclosure, the database 416 may provide a hash table to store the mapping stack 418, in the form of one or more scenic views mapped to the respective one or more poses for the mobile robot 410 for each of mapped coordinate in the environment. In general, the hash table (hash map) is a data structure that implements an associative array abstract data type, a structure that can map keys to values. The hash table uses a hash function to compute an index, also called a hash code, into an array of buckets or slots, from which the desired value can be found; such that during lookup, the key is hashed, and the resulting hash indicates where the corresponding value is stored.

[0052] At step 302, the method 300 includes implementing the mapping stack 418 comprising the one or more scenic views corresponding to respective one or more poses for the mobile robot 410, for each of mapped coordinate in the environment. As may be appreciated by a person skilled in the art, the system 400 may execute a mapping stage beforehand in which the mobile robot 410 itself (including or alternatively other mobile robot(s) from a fleet of mobile robots in the system 400) may be used to generate the mapping stack 418 for the environment. Specifically, the mapping stage may involve synthesis of the mapping stack 418 storing both the images corresponding to the scenic views and the corresponding instant linear/angular coordinate of the mobile robot 410. In some examples, the mobile robot 410 may utilize simultaneous localization and mapping (SLAM) technique to build up the mapping stack of the environment, while at the same time keeping track of its current location and the current pose in the environment. During the initial mapping, when the mobile robot 410 scans the environment to simultaneously create a map of the environment and localize itself within the environment; a simultaneous map of

the monochrome images as captured by the image capturing device 414 of the mobile robot 410 is stored in the mapping stack 418 of the database 416. Herein, the set of images are related to the map coordinates, quantized to a predefined grid. Such mapping stage process may be contemplated by a person of ordinary skill in the art and thus has not been described further for the brevity of the present disclosure.

[0053] Correspondingly, in the present system 400, as shown in FIG. 4, the system 400 includes an access module 402 as part of the processing arrangement 200, to establish communication between the processing arrangement 200 and the database 416. Herein, the access module 402 may provide a medium (e.g., a communication channel) through which the components of the system 400 communicate with each other. Examples of the communication channel may include, but are not limited to, a communication channel in a computer cluster, a Local Area Communication channel (LAN), a cellular communication channel, a wireless sensor communication channel (WSN), a cloud communication channel, a Metropolitan Area Communication channel (MAN), and/or the Internet. Optionally, the access module 402 may include one or more of a wired connection, a wireless network, cellular networks such as 2G, 3G, 4G, 5G mobile networks, and a Zigbee connection.

[0054] Further, the access module 402 is further configured for accessing the mapping stack 418. The access module 402, in the context of present disclosure, provides privilege or assigned permission to access the one or more scenic views of the mapping stack 418, as stored in the database 416. In particular, the access module 402 may allow to selectively access the scenic views based on requirements of navigation of the mobile robot 410 in the environment. For instance, in the present examples, during navigation, the mobile robot 410 may utilize the mapping stack 418 and tries to repeatedly match the current scenic view it sees (based on the current coordinate thereof) with the scenic view it wants to see (for the corresponding coordinate information) from the mapping stack 418. When the expected scenic view may be visible, then the mobile robot 410 may move forward in the direction corresponding to the expected scenic view. Therefore, the mobile robot 410 may need access to the scenic view for the instant coordinate for further navigation, which is provided by the access module 402 herein.

[0055] At step 304, the method 300 includes capturing, by the mobile robot 410, a current scenic view with information about a current pose and a current coordinate thereof, in the

environment. Herein, the processing arrangement 200 may instruct the mobile robot 410 to capture the current scenic view with information about the current pose and the current coordinate thereof, in the environment. The mobile robot 410 may utilize the image capturing device 414 to capture an instant image of the environment in order to determine the current scenic view in a current trajectory of the mobile robot 410. In some examples, the environment, which may be like a factory or warehouse floor, may be divided into grids, with each grid representing one of the coordinates of the environment. For navigation, the mobile robot 410 moves from one coordinate to a next coordinate in the environment, based on the defined trajectory therefor. This is achieved based on the scenic view as visible from the current coordinate and an expected scenic view from the coordinate to which the mobile robot 410 may need to be moved based on the defined trajectory therefor.

[0056] At step 306, the method 300 includes comparing the current scenic view with the scenic view from the mapping stack 418 having the corresponding pose and the corresponding mapped coordinate matching the current pose and the current coordinate. Herein, the processing arrangement 200 is configured to compare the current scenic view with the scenic view from the mapping stack 418 having the corresponding pose and the corresponding mapped coordinate matching the current pose and the current coordinate. As illustrated in FIG. 4, the processing arrangement 200 may include an image processing module 404 for performing comparison between the two images, i.e., the current scenic view with the scenic view from the mapping stack 418. Such image processing module 404 may implement any of known suitable image comparison algorithm for the present purpose, including, but not limited to, strict comparison technique, fuzzy pixel comparison technique, histogram comparison technique, correlation comparison technique, and the like.

[0057] As discussed, during navigation, the mobile robot 410 may utilize the mapping stack 418 and tries to repeatedly match the current scenic view it sees (based on the current coordinate thereof) with the scenic view it wants to see (for the corresponding coordinate information) from the mapping stack 418. When the captured scenic view may match the expected scenic view, then the mobile robot 410 may move forward in the direction corresponding to the expected scenic view. It may be appreciated that the pose of the mobile robot 410 may be considered while matching the scenic view from the mapping stack 418 and the captured scenic view. During the navigation stage, the current localization argument for the mobile robot 410

(including the linear pose and the rotational pose) is used to isolate the quantized linear and angular coordinates. The quantized linear and angular coordinates map to the images stored during the mapping stage. The set of saved images is compared with the live image as captured by the image capturing device 414 of the mobile robot 410 (after rejecting disturbances).

[0058] Now if there is a change in discrete features in the environment and/or the presence of slow moving objects in the environment, the mobile robot 410 may not be able to find a decent match for the current scenic view and thus may not be able to localize itself in the environment. The localization issue due to slow moving objects in the environment or the change in discrete features in the environment lead to drift based errors in robot localization. Such errors in localization might lead to catastrophic accidents as the mobile robot may hit other objects. Therefore, it is desirable to describe a need to remap at intermediate time instances such that the error in subsequent localization and mapping is mitigated.

[0059] At step 308, the method 300 includes calculating a matching score for the current scenic view based on the comparison. Herein, the processing arrangement 200 is configured to calculate the matching score for the current scenic view based on the comparison. As illustrated in FIG. 4, the processing arrangement 200 may include an analysis module 406 for performing the necessary mathematical calculations to determine the matching score. As used herein, the matching score may represent a degree of match of the current scenic view with the scenic view from the mapping stack 418 having the corresponding pose and the corresponding mapped coordinate matching the current pose and the current coordinate. Specifically, in the present embodiments, the matching score is generated for each pose of the mobile robot 410 for its current coordinate. That is, the matching scores are generated for all the points which the mobile robot 410 claims to have localized at.

[0060] For this purpose, the method 300 includes capturing, by the mobile robot 410, a series of current scenic views with information about the corresponding current poses and the corresponding current coordinates, over a trajectory of the mobile robot 410 in the environment. Herein, the processing arrangement 200 is configured to instruct the mobile robot 410 to capture the series of current scenic views with information about the corresponding current poses and the corresponding current coordinates, over a trajectory of the mobile robot 410 in the environment. The method 300 further includes comparing each of the series of current scenic views with the

correlative scenic view from the mapping stack 418, having the corresponding pose and the corresponding mapped coordinate matching the respective current pose and the respective current coordinate. Herein, the image processing module 404 in the processing arrangement 200 is configured to compare each of the series of current scenic views with the correlative scenic view from the mapping stack 418, having the corresponding pose and the corresponding mapped coordinate matching the respective current pose and the respective current coordinate. The method 300 further includes determining an individual matching score for each of the series of current scenic views based on each of the comparison. Herein, the analysis module 406 in the processing arrangement 200 is configured to determine the individual matching score for each of the series of current scenic views based on each of the comparison.

[0061] The method 300 further includes calculating the matching score as a discounted sum of the individual matching scores. Herein, the analysis module 406 in the processing arrangement 200 is configured to calculate the matching score as the discounted sum of the individual matching scores. The discounted sum of the individual matching scores represents a discounted sum of rewards for the trajectory which the mobile robot 410 has visited. In the present embodiments, the matching score as the discounted sum of the individual matching scores is based on a weighted average of the individual matching scores, with a later of the current scenic view of the series of current scenic views being given a higher weight in comparison to former of the current scenic view of the series of current scenic views. It may be appreciated that herein the weighted average is being used so that any one image frame (scenic view) which may have an obstacle, such as a human passing by a front of the mobile robot 410, may not substantially skew the overall matching score. Further, the instant (live) image frame may be given higher weightage as compared to previously captured images, as that may have more relevance to immediate action of the mobile robot 410. Further, the matching score as the discounted sum of the individual matching scores is based on a rate of reduction of the individual matching scores from the former of the current scenic view of the series of current scenic views to the later of the current scenic view of the series of current scenic views being substantially constant in relation to the series of current scenic views. Again, herein, the rate of reduction of the individual matching scores is preferred to be substantially constant any one image frame (scenic view) which may have an obstacle, such as a human passing by a front of the mobile robot 410, may not substantially skew the overall matching score.

[0062] At step 310, the method 300 further includes generating a localization error flag in case of the matching score being below a predefined threshold. Herein, the processing arrangement 200 is configured to generate the localization error flag in case of the matching score being below the predefined threshold. As illustrated in FIG. 4, the processing arrangement 200 may include an error flagging module 408 to compare the matching score with the predefined threshold. The error flagging module 408 may generate the localization error flag if the matching score is determined to be below the predefined threshold in the said comparison. The predefined threshold may be specified based on observation of errors in localization of the mobile robot 410, as would be contemplated. The localization error flag may be in the form of a signal or an alarm indicative of error in localization of the mobile robot 410. Further, the localization error flag is in turn indicative of a need to remap the environment such that the error in subsequent localization and mapping is mitigated.

[0063] The present embodiments provide that there is a need to reconfirm whether the generated localization error flag may be discounted. In one or more embodiments, the method 300 includes configuring the mobile robot 410 to perform one or more secondary manoeuvres, in case of generation of the localization error flag. Herein, the processing arrangement 200 is configured to instruct the mobile robot 410 to perform one or more secondary manoeuvres, in case of generation of the localization error flag. Herein, the secondary manoeuvres may be in the form of manual inspection of the mobile robot (such as, for re-localization therefor), instruction for the mobile robot 410 to reach a nearest ground marker, configuring the mobile robot 410 to use one or more other mobile robots as markers for re-localization therefor. The method 300 further includes confirming a localization error in the navigation of the mobile robot 410 based on the one or more secondary manoeuvres. Herein, the processing arrangement 200 is configured to confirm the localization error in the navigation of the mobile robot 410 based on the one or more secondary manoeuvres. That is, even after performing the secondary manoeuvres, the mobile robot 410 may not be able to localize itself in the environment, then it may be confirmed that there is the localization error in the navigation of the mobile robot 410.

[0064] Once the localization error in the navigation of the mobile robot 410 may be confirmed, the method 300 and the system 400 of the present disclosure may perform the necessary steps to remap the environment. In one or more embodiments, the method 300 includes appending the current scenic view with information about the current pose and the current coordinate thereof

in the mapping stack 418, based on confirmation of the localization error in the navigation of the mobile robot 410. Herein, the processing arrangement 200 is configured to append the current scenic view with information about the current pose and the current coordinate thereof in the mapping stack 418 stored in the database 416, based on confirmation of the localization error in the navigation of the mobile robot 410. In particular, the mobile robot 410 may perform the predefined secondary manoeuvre, and restarts the matching pipeline. If, again, the discounted sum is below the predefined threshold, the mobile robot 410 is considered to be lost, and the localization error flag is generated. If the localization error flag for the mobile robot 410 is not raised, then the modified set of images (with the current scenic view including information about the current pose and the current coordinate of the mobile robot 410) is appended to the mapping stack 418 in the database 416 (which led to the localization error flag). The modification to the database 416 may either be direct addition of new images, modification of the previous images to inculcate the changes or a combination of the two.

[0065] FIG. 5 illustrates a process flow (represented generally by reference numeral 500) for determination of localization error in the navigation of the mobile robot 410 in the environment (represented in the form of a grid of coordinates with reference numeral 502), in accordance with one or more embodiments of the present disclosure. As may be seen, based on a query (represented by a block 510) about pose and orientation trajectory of the mobile robot 410, a series of images (scenic views), generally represented as a "set" with Image 1, Image 2, and Image 3 in FIG. 5, may be captured with information about the corresponding current poses and the corresponding current coordinates, over a trajectory of the mobile robot 410 in the environment 502. Thereby, the synthesized map may include information about the scenic view the mobile robot 410 sees while in a particular orientation at a pose. When queried (as in block 510) about the desired image that is, the image the mobile robot 410 sees, a matching score is calculated (represented by a block 520) based on the current scene (matched features) and imaged scene trajectories. As discussed, the matching score is calculated as a discounted sum of the individual matching scores for each of the series of current scenic views. Then the matching score is used to generate the localization error flag (represented by a block 520) in case of the matching score being below the predefined threshold, which in turn may indicate a need to remap the environment 502.

[0066] The present disclosure focuses on detecting localization errors in the mapping stack (like a standard localization map) either due to a change in the environment or an error in the

map. During the mapping stage, a simultaneous fully connected graph is developed which corresponds to the scenic view vis-a-vis the robot pose. Once the mapping stack, including the coordinate map and the scenic map being simultaneously created and stored in a hash table of a database, is implemented; the mobile robot tries to repeatedly match the scenic view it sees, and it wants to see (from the stored hash table). The scenes are generated from the stored hash table (unordered map) based on the current coordinates of the mobile robot. Thereafter, the matching score is generated over a trajectory of sequences as a discounted sum of individual matching scores over the defined trajectory of the mobile robot. If the matching score is less than the predefined threshold, a preliminary localization error flag is generated (raised). Then the mobile robot performs a set of secondary manoeuvres to confirm whether the mobile robot is actually lost. If confirmed, an erroneous primary flag is generated, and then the mobile robot appends the new features in the previously saved mapping stack. The localization error flag is useful as it reduces the undesired behaviour of the mobile robot and cancels the probability of accidents, i.e., to prevent further false localization errors. Thus, the present disclosure allows the mobile robot to raise error to avoid undefined behaviour, thereby preventing accidents and improving the quality of localization. Thereby, the present disclosure may be implemented to develop remapping routines in a slow changing environment. Remapping allows the mobile robot to regain localization accuracy, thereby leading to precise navigation.

[0067] The foregoing descriptions of specific embodiments of the present disclosure have been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the present disclosure to the precise forms disclosed, and obviously many modifications and variations are possible in light of the above teaching. The exemplary embodiment was chosen and described in order to best explain the principles of the present disclosure and its practical application, to thereby enable others skilled in the art to best utilize the present disclosure and various embodiments with various modifications as are suited to the particular use contemplated.

We Claim:

1. A method for determining an error in localization in navigation of a mobile robot in an environment, the method comprising:

implementing a mapping stack comprising one or more scenic views corresponding to respective one or more poses for the mobile robot, for each of mapped coordinate in the environment;

capturing, by the mobile robot, a current scenic view with information about a current pose and a current coordinate thereof, in the environment;

comparing the current scenic view with the scenic view from the mapping stack having the corresponding pose and the corresponding mapped coordinate matching the current pose and the current coordinate;

calculating a matching score for the current scenic view based on the comparison; and

generating a localization error flag in case of the matching score being below a predefined threshold.

2. The method as claimed in claim 1 further comprising:

configuring the mobile robot to perform one or more secondary manoeuvres, in case of generation of the localization error flag; and

confirming a localization error in the navigation of the mobile robot based on the one or more secondary manoeuvres.

3. The method as claimed in claim 2 further comprising appending the current scenic view with information about the current pose and the current coordinate thereof in the mapping stack, based on confirmation of the localization error in the navigation of the mobile robot.

4. The method as claimed in claim 1 further comprising:

capturing, by the mobile robot, a series of current scenic views with information about the corresponding current poses and the corresponding current coordinates, over a trajectory of the mobile robot in the environment;

comparing each of the series of current scenic views with the correlative scenic view from the mapping stack, having the corresponding pose and the corresponding mapped coordinate matching the respective current pose and the respective current coordinate;

determining an individual matching score for each of the series of current scenic views based on each of the comparison; and

calculating the matching score as a discounted sum of the individual matching scores.

5. The method as claimed in claim 4, wherein calculating the matching score as the discounted sum of the individual matching scores is based on:

a weighted average of the individual matching scores, with a later of the current scenic view of the series of current scenic views being given a higher weight in comparison to former of the current scenic view of the series of current scenic views; and

a rate of reduction of the individual matching scores from the former of the current scenic view of the series of current scenic views to the later of the current scenic view of the series of current scenic views being substantially constant in relation to the series of current scenic views.

6. A system for determining an error in localization in navigation of a mobile robot in an environment, the system comprising:

a database having stored a mapping stack comprising one or more scenic views corresponding to respective one or more poses for the mobile robot, for each of mapped coordinate in the environment; and

a processing arrangement in communication with the database to access the mapping stack, the processing arrangement configured to:

instruct the mobile robot to capture a current scenic view with information about a current pose and a current coordinate thereof, in the environment;

compare the current scenic view with the scenic view from the mapping stack having the corresponding pose and the corresponding mapped coordinate matching the current pose and the current coordinate;

calculate a matching score for the current scenic view based on the comparison; and

generate a localization error flag in case of the matching score being below a predefined threshold.

7. The system as claimed in claim 6, wherein the processing arrangement is further configured to:

instruct the mobile robot to perform one or more secondary manoeuvres, in case of generation of the localization error flag; and

confirm a localization error in the navigation of the mobile robot based on the one or more secondary manoeuvres.

8. The system as claimed in claim 7, wherein the processing arrangement is further configured to append the current scenic view with information about the current pose and the current coordinate thereof in the mapping stack stored in the database, based on confirmation of the localization error in the navigation of the mobile robot.

9. The system as claimed in claim 6, wherein the processing arrangement is further configured to:

instruct the mobile robot to capture a series of current scenic views with information about the corresponding current poses and the corresponding current coordinates, over a trajectory of the mobile robot in the environment;

compare each of the series of current scenic views with the correlative scenic view from the mapping stack, having the corresponding pose and the corresponding mapped coordinate matching the respective current pose and the respective current coordinate;

determine an individual matching score for each of the series of current scenic views based on each of the comparison; and

calculate the matching score as a discounted sum of the individual matching scores.

10. The system as claimed in claim 9, wherein calculating the matching score as the discounted sum of the individual matching scores is based on:

a weighted average of the individual matching scores, with a later of the current scenic view of the series of current scenic views being given a higher weight in comparison to former of the current scenic view of the series of current scenic views; and

a rate of reduction of the individual matching scores from the former of the current scenic view of the series of current scenic views to the later of the current scenic view of the series of current scenic views being substantially constant in relation to the series of current scenic views.

Dated this 29th Day of March, 2022

- Digitally signed-

Sujit Thakur
IN/PA-1124
Agent for the Applicant

ABSTRACT

SYSTEM AND METHOD FOR DETERMINING ERROR IN LOCALIZATION IN NAVIGATION OF MOBILE ROBOT

A system and method for determining an error in localization in navigation of a mobile robot in an environment. The method comprises implementing a mapping stack comprising one or more scenic views corresponding to respective one or more poses for the mobile robot, for each of mapped coordinate in the environment. The method further comprises capturing, by the mobile robot, a current scenic view with information about a current pose and a current coordinate thereof, in the environment. The method further comprises comparing the current scenic view with the scenic view from the mapping stack having the corresponding pose and the corresponding mapped coordinate matching the current pose and the current coordinate. The method further comprises calculating a matching score for the current scenic view based on the comparison. The method further comprises generating a localization error flag in case of the matching score being below a predefined threshold.

FIG. 3

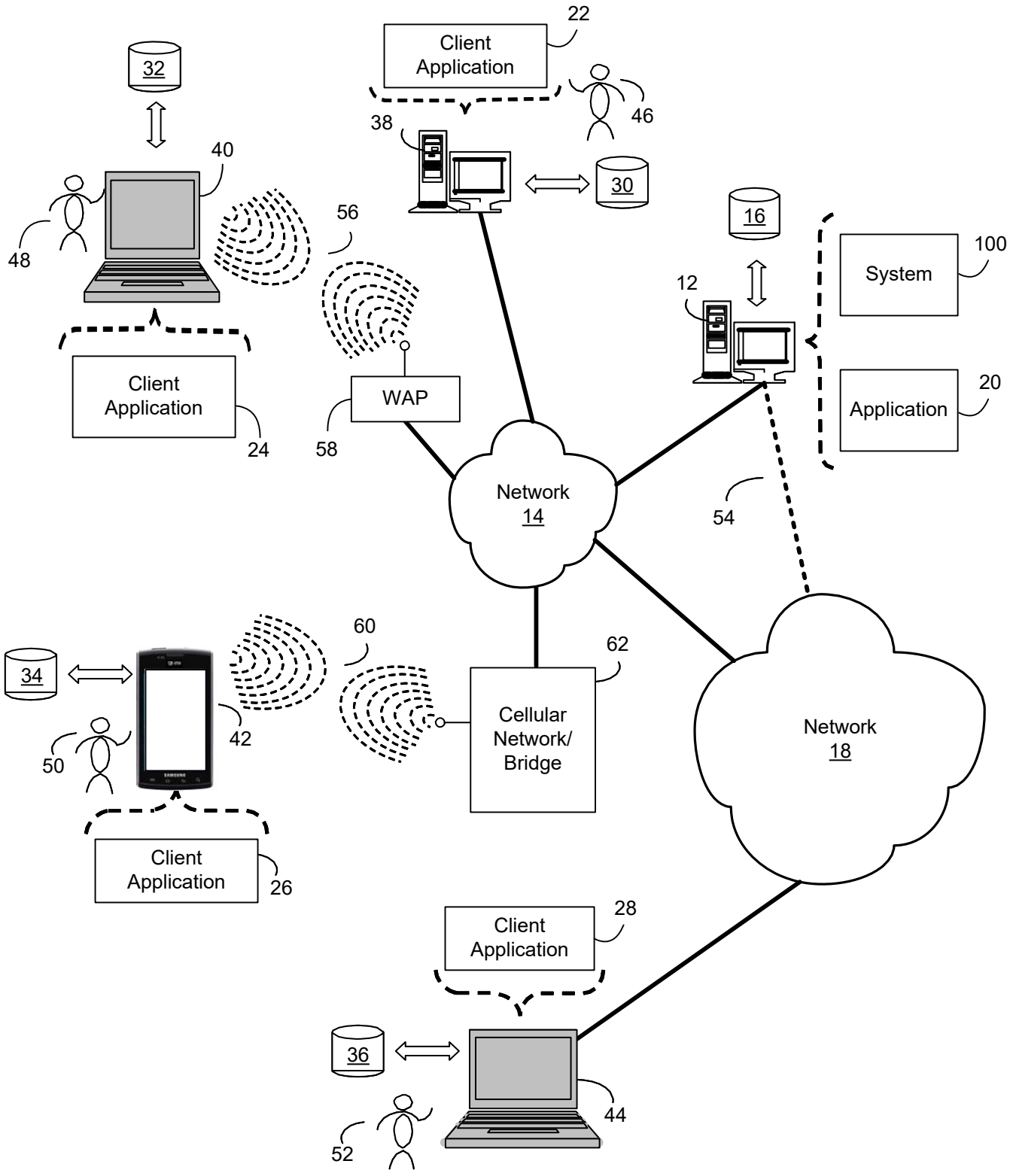

FIG. 1

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant

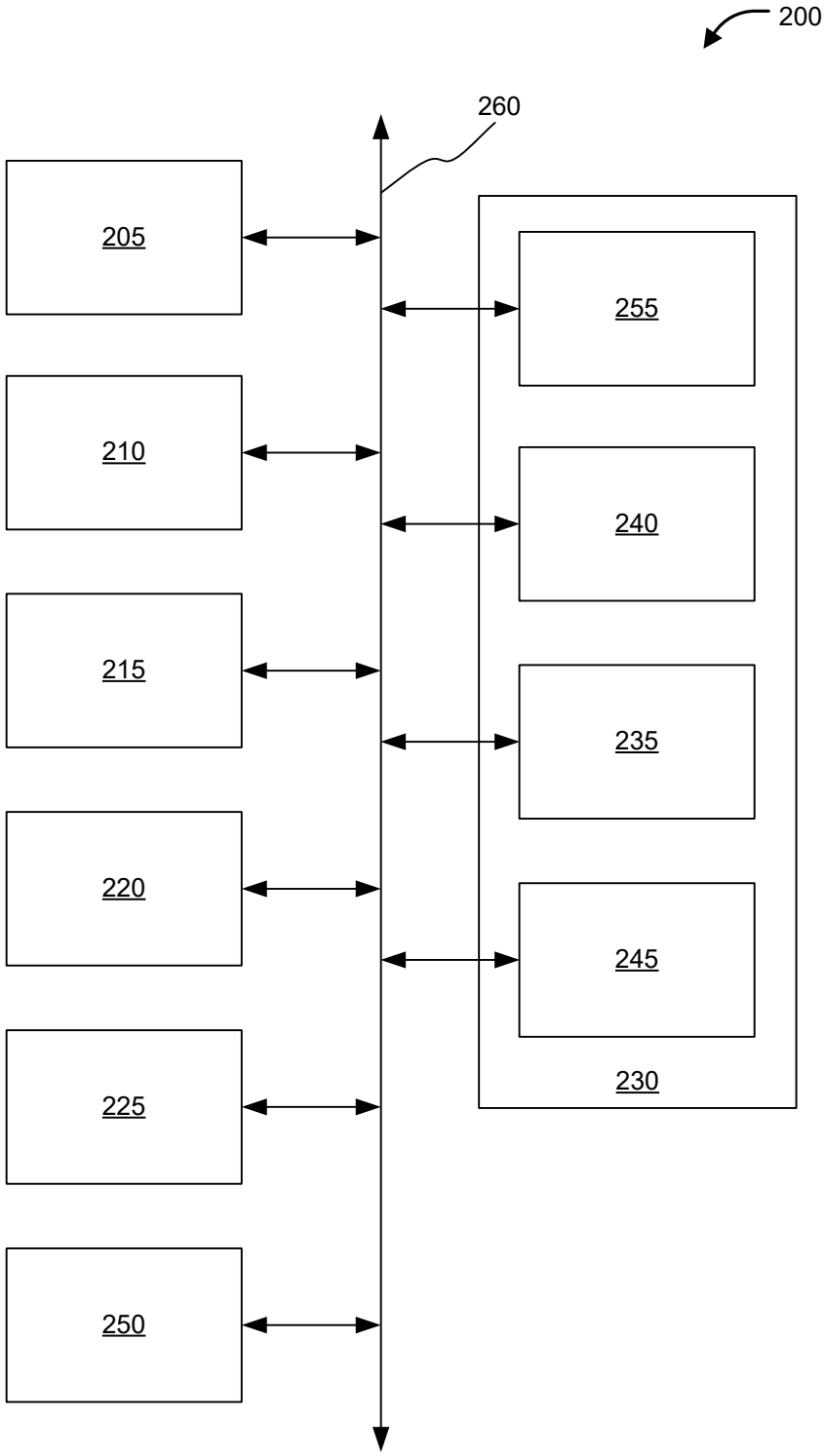

FIG. 2

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant

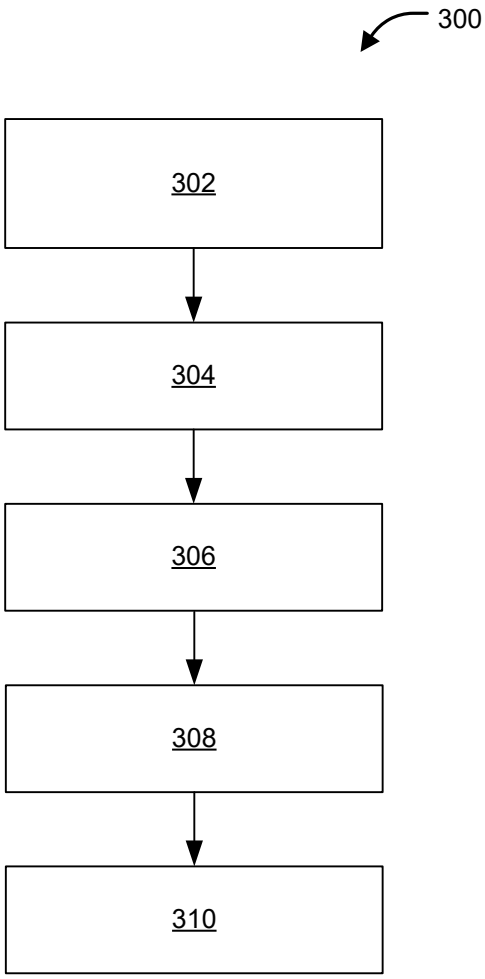

FIG. 3

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant

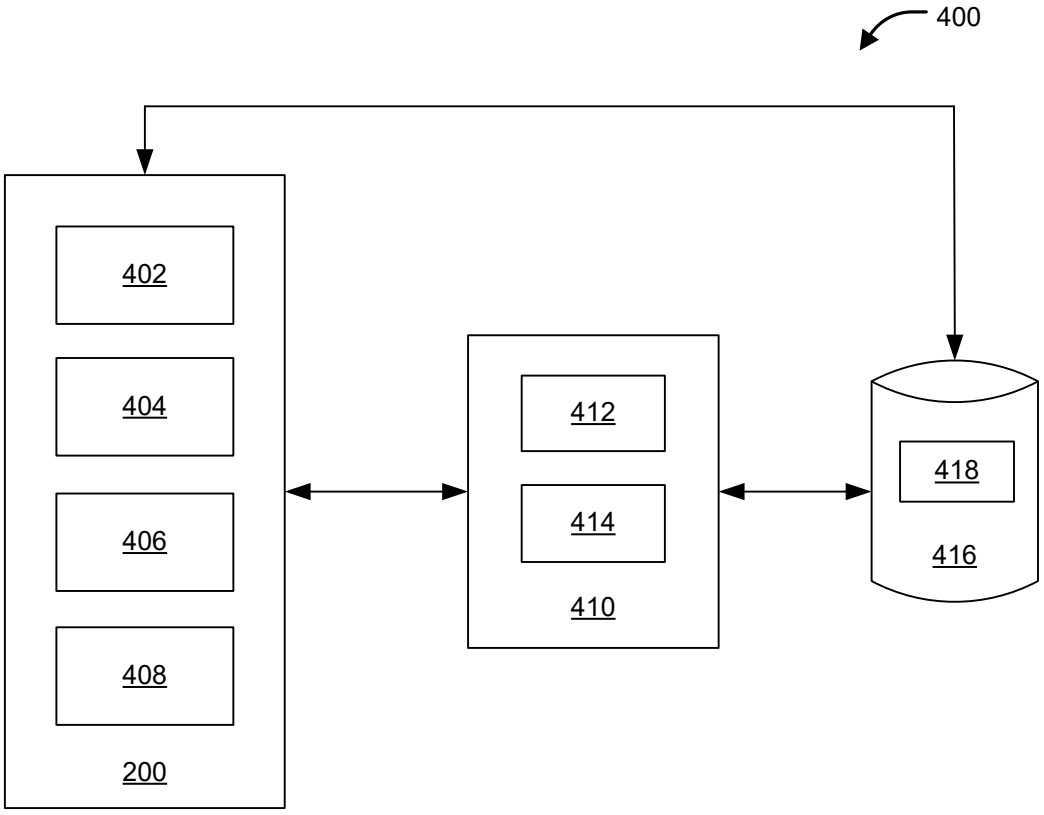

FIG. 4

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant

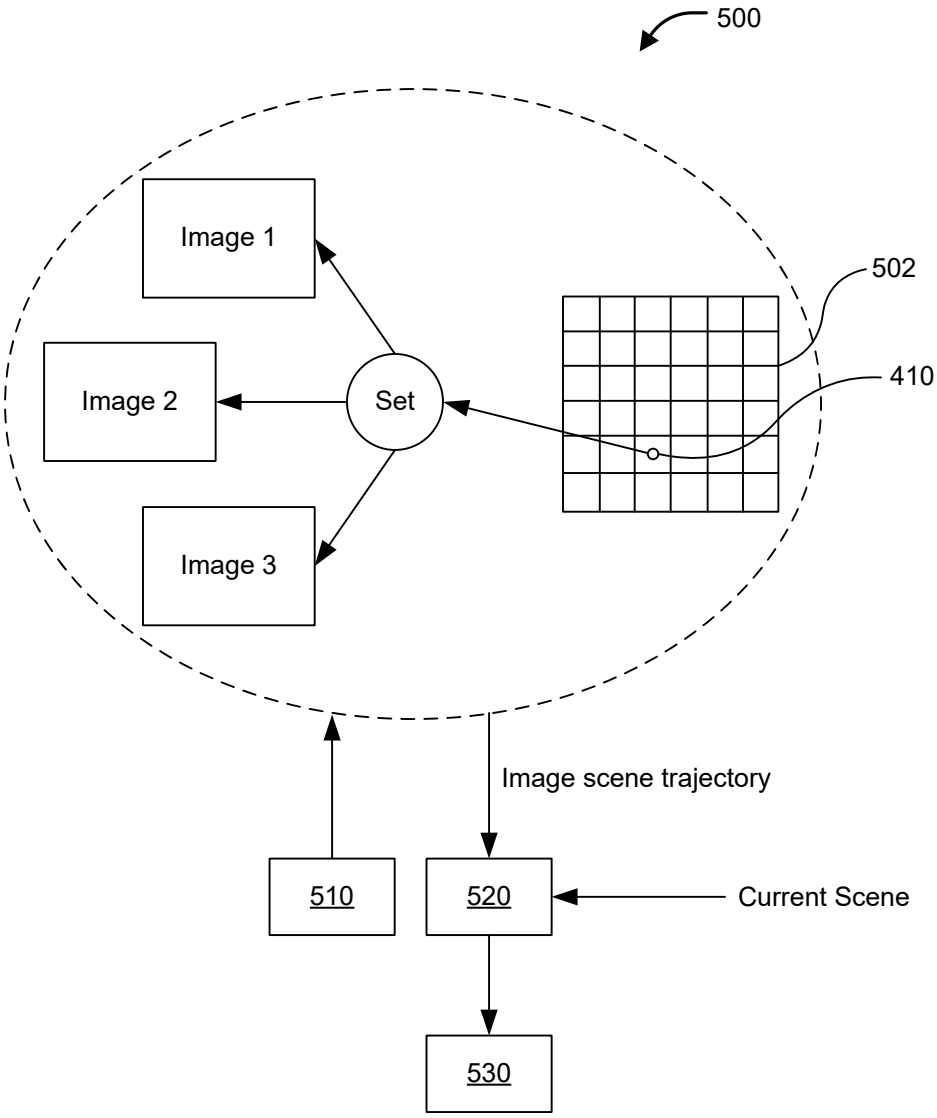

FIG. 5

- Digitally signed-
Sujit Thakur
IN/PA-1124
Agent for the Applicant