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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 December 2020 (10.12.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/247066 A1

(51) International Patent Classification:
B25J 9/16 (2006.01)

(21) International Application Number:
PCT/US2020/027948

(22) International Filing Date:
13 April 2020 (13.04.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
16/429,906 03 June 2019 (03.06.2019) US

(71) Applicant: **ZEBRA TECHNOLOGIES CORPORATION** [US/US]; 3 Overlook Point, Lincolnshire, Illinois 60069 (US).

(72) Inventors: **SAKHDARI, Bijan**; 350 Columbia Street West, Unit 162, Waterloo, Ontario N2L 6P4 (CA). **TAJEDDIN, Sadegh**; 1423 Mississauga Valley Blvd., Apt 1207, Mississauga, Ontario L5N 7J9 (CA). **GUO, Zi Cong**; 3571 Havenwood Drive, Mississauga, Ontario L4X 2N1 (CA).

(74) Agent: **KAPMAR, Dimitry** et al.; 3 Overlook Point, Lincolnshire, Illinois 60069 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

(54) **Title:** METHOD, SYSTEM AND APPARATUS FOR DYNAMIC TASK SEQUENCING

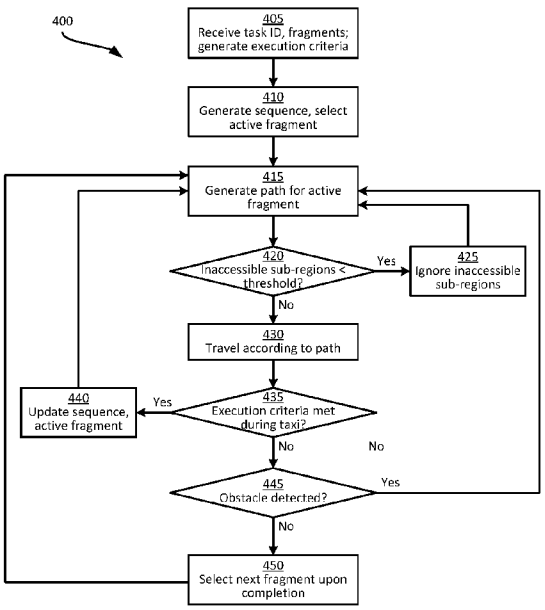


FIG. 4

(57) **Abstract:** A method in a navigational controller includes: obtaining (i) a plurality of task fragments identifying respective sets of sub-regions in a facility, and (ii) an identifier of a task to be performed by a mobile automation apparatus at each of the sets of sub-regions; selecting an active one of the task fragments according to a sequence specifying an order of execution of the task fragments; generating a path including (i) a taxi portion from a current position of the mobile automation apparatus to the sub-regions identified by the active task fragment, and (ii) an execution portion traversing the sub-regions identified by the active task fragment; during travel along the taxi portion, determining, based on a current pose of the mobile automation apparatus, whether to initiate execution of another task fragment; and when the determination is affirmative, updating the sequence to mark the other task fragment as the active task fragment.

WO 2020/247066 A1

[Continued on next page]

Published:
— *with international search report (Art. 21(3))*

METHOD, SYSTEM AND APPARATUS FOR DYNAMIC TASK SEQUENCING

BACKGROUND

[0001] Environments in which objects are managed, such as retail facilities, warehousing and distribution facilities, and the like, may be complex and fluid. For example, a retail facility may include objects such as products for purchase, and a distribution facility may include objects such as parcels or pallets. Each facility may also contain dynamic obstacles such as people, vehicles and the like. A mobile apparatus may be deployed within such facilities to perform tasks at various locations. For example, a mobile apparatus may be deployed to capture data at various locations in a retail facility. Determining a path for the mobile apparatus to travel efficiently among the above-mentioned locations, however, may be computationally intensive.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0002] The accompanying figures, where like reference numerals refer to identical or functionally similar elements throughout the separate views, together with the detailed description below, are incorporated in and form part of the specification, and serve to further illustrate embodiments of concepts that include the claimed invention, and explain various principles and advantages of those embodiments.

[0003] FIG. 1 is a schematic of a mobile automation system.

[0004] FIG. 2 depicts a mobile automation apparatus in the system of FIG. 1.

[0005] FIG. 3 is a block diagram of certain internal components of the mobile automation apparatus in the system of FIG. 1.

[0006] FIG. 4 is a flowchart of a task sequencing method in the system of FIG. 1.

[0007] FIG. 5 is a diagram of a retail facility illustrating the generation of execution criteria at block 405 of the method of FIG. 4.

[0008] FIG. 6 is a diagram illustrating an example performance of block 415 of the method of FIG. 4.

[0009] FIG. 7 is a diagram illustrating a further example performance of block 415 of the method of FIG. 4.

[0010] FIG. 8 is a diagram illustrating an example performance of block 435 of the method of FIG. 4.

[0011] FIG. 9 is a diagram illustrating an example performance of block 440 of the method of FIG. 4.

[0012] FIG. 10 is a diagram illustrating an example performance of blocks 415 and 435 of the method of FIG. 4.

[0013] FIG. 11 is a diagram illustrating an example performance of blocks 420 and 425 of the method of FIG. 4.

[0014] FIG. 12 is a diagram illustrating another example performance of blocks 420 and 425 of the method of FIG. 4.

[0015] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been drawn to scale. For example, the dimensions of some of the elements in the figures may be exaggerated relative to other elements to help to improve understanding of embodiments of the present invention.

[0016] The apparatus and method components have been represented where appropriate by conventional symbols in the drawings, showing only those specific details that are pertinent to understanding the embodiments of the present invention so as not to obscure the disclosure with details that will be readily apparent to those of ordinary skill in the art having the benefit of the description herein.

DETAILED DESCRIPTION

[0017] Examples disclosed herein are directed to a method in a navigational controller, including: obtaining (i) a plurality of task fragments identifying respective sets of sub-regions in a facility, and (ii) an identifier of a task to be performed by a mobile automation apparatus at each of the sets of sub-regions; selecting an active one of the task fragments according to a sequence specifying an order of execution of the task fragments; generating a path including (i) a taxi portion from a current position of the mobile automation apparatus to the sub-regions identified by the active task fragment, and (ii) an execution portion traversing the sub-regions identified by the active task fragment; during travel along the taxi portion, determining, based on a current pose of the mobile automation apparatus, whether to initiate execution of another task

fragment; and when the determination is affirmative, updating the sequence to mark the other task fragment as the active task fragment.

[0018] Additional examples disclosed herein are directed to a mobile automation apparatus, comprising: a locomotive assembly; a data capture sensor; and a navigational controller configured to: obtain (i) a plurality of task fragments identifying respective sets of sub-regions in a facility, and (ii) an identifier of a task to be performed at each of the sets of sub-regions; select an active one of the task fragments according to a sequence specifying an order of execution of the task fragments; generate a path including (i) a taxi portion from a current position of the mobile automation apparatus to the sub-regions identified by the active task fragment, and (ii) an execution portion traversing the sub-regions identified by the active task fragment; control the locomotive assembly to travel along the path; during travel along the taxi portion, determine, based on a current pose of the mobile automation apparatus, whether to initiate execution of another task fragment; and when the determination is affirmative, update the sequence to mark the other task fragment as the active task fragment.

[0019] Further examples disclosed herein are directed to a method in a navigational controller comprising: obtaining a sequence of task fragments identifying respective sets of sub-regions in a facility; selecting an active one of the task fragments according to the sequence; controlling a mobile automation apparatus to perform a task at the set of sub-regions identified by the active task fragment; responsive to a change in a pose of the mobile automation apparatus during performance of the task, updating the sequence to mark another task fragment as the active task fragment.

[0020] FIG. 1 depicts a mobile automation system 100 in accordance with the teachings of this disclosure. The system 100 includes a server 101 in communication with at least one mobile automation apparatus 103 (also referred to herein simply as the apparatus 103) and at least one client computing device 104 via communication links 105, illustrated in the present example as including wireless links. In the present example, the links 105 are provided by a wireless local area network (WLAN) deployed via one or more access points (not shown). In other examples, the server 101, the client device 104, or both, are located remotely (i.e. outside the environment in which the apparatus 103 is deployed), and the links 105 therefore include wide-area networks such as the

Internet, mobile networks, and the like. The system 100 also includes a dock 106 for the apparatus 103 in the present example. The dock 106 is in communication with the server 101 via a link 107 that in the present example is a wired link. In other examples, however, the link 107 is a wireless link.

[0021] The client computing device 104 is illustrated in FIG. 1 as a mobile computing device, such as a tablet, smart phone or the like. In other examples, the client device 104 is implemented as another type of computing device, such as a desktop computer, a laptop computer, another server, a kiosk, a monitor, and the like. The system 100 can include a plurality of client devices 104 in communication with the server 101 via respective links 105.

[0022] The system 100 is deployed, in the illustrated example, in a retail facility including a plurality of support structures such as shelf modules 110-1, 110-2, 110-3 and so on (collectively referred to as shelf modules 110 or shelves 110, and generically referred to as a shelf module 110 or shelf 110 – this nomenclature is also employed for other elements discussed herein). Each shelf module 110 supports a plurality of products 112. Each shelf module 110 includes a shelf back 116-1, 116-2, 116-3 and a support surface (e.g. support surface 117-3 as illustrated in FIG. 1) extending from the shelf back 116 to a shelf edge 118-1, 118-2, 118-3.

[0023] The shelf modules 110 are typically arranged in a plurality of aisles, each of which includes a plurality of modules 110 aligned end-to-end. In such arrangements, the shelf edges 118 face into the aisles, through which customers in the retail facility as well as the apparatus 103 may travel. As will be apparent from FIG. 1, the term "shelf edge" 118 as employed herein, which may also be referred to as the edge of a support surface (e.g., the support surfaces 117) refers to a surface bounded by adjacent surfaces having different angles of inclination. In the example illustrated in FIG. 1, the shelf edge 118-3 is at an angle of about ninety degrees relative to each of the support surface 117-3 and the underside (not shown) of the support surface 117-3. In other examples, the angles between the shelf edge 118-3 and the adjacent surfaces, such as the support surface 117-3, is more or less than ninety degrees.

[0024] The apparatus 103 is equipped with a plurality of navigation and data capture sensors 108, such as image sensors (e.g. one or more digital cameras) and depth sensors

(e.g. one or more Light Detection and Ranging (LIDAR) sensors, one or more depth cameras employing structured light patterns, such as infrared light, or the like). The apparatus 103 is deployed within the retail facility and, via communication with the server 101 and use of the sensors 108, navigates autonomously or partially autonomously along a length 119 of at least a portion of the shelves 110. Navigation may be performed according to a frame of reference 102 established within the retail facility. That is, the apparatus 103 tracks its location in the frame of reference 102.

[0025] While navigating among the shelves 110, the apparatus 103 can capture images, depth measurements and the like, representing the shelves 110 (generally referred to as shelf data or captured data). As will be discussed in greater detail below, the apparatus 103 receives instructions (e.g. from the server 101) to perform a task such as the above-mentioned data collection with respect to a plurality of shelf modules 110. In addition to navigating among the shelves 110 to perform the specified task, the apparatus 103 can also determine and update a sequence in which the shelf modules 110 are traversed for data capture. Determining and updating the sequence enables the apparatus 103 to traverse the shelf modules 110 in an order that favors travel time spent conducting data capture over travel time spent taxiing (i.e. travelling without capturing data).

[0026] The server 101 includes a special purpose controller, such as a processor 120, specifically designed to control and/or assist the mobile automation apparatus 103 to navigate the environment and to capture data. The processor 120 is interconnected with a non-transitory computer readable storage medium, such as a memory 122, having stored thereon computer readable instructions for performing various functionality, including control of the apparatus 103 to navigate the modules 110 and capture shelf data, as well as post-processing of the shelf data. The memory 122 can also store data for use in the above-mentioned control of the apparatus 103, such as a repository 123 containing a map of the retail environment and any other suitable data (e.g. operational constraints used to control the apparatus 103, data captured by the apparatus 103, and the like).

[0027] The memory 122 includes a combination of volatile memory (e.g. Random Access Memory or RAM) and non-volatile memory (e.g. read only memory or ROM, Electrically Erasable Programmable Read Only Memory or EEPROM, flash memory).

The processor 120 and the memory 122 each comprise one or more integrated circuits. In some embodiments, the processor 120 is implemented as one or more central processing units (CPUs) and/or graphics processing units (GPUs).

[0028] The server 101 also includes a communications interface 124 interconnected with the processor 120. The communications interface 124 includes suitable hardware (e.g. transmitters, receivers, network interface controllers and the like) allowing the server 101 to communicate with other computing devices – particularly the apparatus 103, the client device 104 and the dock 106 – via the links 105 and 107. The links 105 and 107 may be direct links, or links that traverse one or more networks, including both local and wide-area networks. The specific components of the communications interface 124 are selected based on the type of network or other links that the server 101 is required to communicate over. In the present example, as noted earlier, a wireless local-area network is implemented within the retail facility via the deployment of one or more wireless access points. The links 105 therefore include either or both wireless links between the apparatus 103 and the mobile device 104 and the above-mentioned access points, and a wired link (e.g. an Ethernet-based link) between the server 101 and the access point.

[0029] The processor 120 can therefore obtain data captured by the apparatus 103 via the communications interface 124 for storage (e.g. in the repository 123) and subsequent processing (e.g. to detect objects such as shelved products in the captured data, and detect status information corresponding to the objects). The server 101 may also transmit status notifications (e.g. notifications indicating that products are out-of-stock, in low stock or misplaced) to the client device 104 responsive to the determination of product status data. The client device 104 includes one or more controllers (e.g. central processing units (CPUs) and/or field-programmable gate arrays (FPGAs) and the like) configured to process (e.g. to display) notifications received from the server 101.

[0030] Turning now to FIG. 2, the mobile automation apparatus 103 is shown in greater detail. The apparatus 103 includes a chassis 201 containing a locomotive mechanism 203 (e.g. one or more electrical motors driving wheels, tracks or the like). The apparatus 103 further includes a sensor mast 205 supported on the chassis 201 and, in the present

example, extending upwards (e.g., substantially vertically) from the chassis 201. The mast 205 supports the sensors 108 mentioned earlier. In particular, the sensors 108 include at least one imaging sensor 207, such as a digital camera, as well as at least one depth sensor 209, such as a 3D digital camera capable of capturing both depth data and image data. The apparatus 103 also includes additional depth sensors, such as LIDAR sensors 211. As shown in FIG. 2, the cameras 207 and the LIDAR sensors 211 are arranged on one side of the mast 205, while the depth sensor 209 is arranged on a front of the mast 205. That is, the depth sensor 209 is forward-facing (i.e. captures data in the direction of travel of the apparatus 103), while the cameras 207 and LIDAR sensors 211 are side-facing (i.e. capture data alongside the apparatus 103, in a direction perpendicular to the direction of travel). In other examples, the apparatus 103 includes additional sensors, such as one or more RFID readers, temperature sensors, and the like.

[0031] In the present example, the mast 205 supports seven digital cameras 207-1 through 207-7, and two LIDAR sensors 211-1 and 211-2. The mast 205 also supports a plurality of illumination assemblies 213, configured to illuminate the fields of view of the respective cameras 207. That is, the illumination assembly 213-1 illuminates the field of view of the camera 207-1, and so on. The sensors 207 and 211 are oriented on the mast 205 such that the fields of view of each sensor face a shelf 110 along the length 119 of which the apparatus 103 is traveling. The apparatus 103 is configured to track a location of the apparatus 103 (e.g. a location of the center of the chassis 201) in a common frame of reference previously established in the retail facility, permitting data captured by the mobile automation apparatus to be registered to the frame of reference.

[0032] Referring to FIG. 3, certain components of the mobile automation apparatus 103 are shown, in addition to the cameras 207, depth sensor 209, lidars 211, and illumination assemblies 213 mentioned above. The apparatus 103 includes a special-purpose controller, such as a processor 300, interconnected with a non-transitory computer readable storage medium, such as a memory 304. The memory 304 includes a suitable combination of volatile memory (e.g. Random Access Memory or RAM) and non-volatile memory (e.g. read only memory or ROM, Electrically Erasable Programmable Read Only Memory or EEPROM, flash memory). The processor 300 and the memory 304 each comprise one or more integrated circuits. The memory 304 stores computer readable instructions for execution by the processor 300. In particular, the memory 304

stores a navigational application 308 which, when executed by the processor 300, configures the processor 300 to perform various functions discussed below in greater detail and related to the navigation of the apparatus 103 (e.g. by controlling the locomotive mechanism 203). The processor 300, when so configured by the execution of the application 308, may also be referred to as a navigational controller 300. Those skilled in the art will appreciate that the functionality implemented by the processor 300 via the execution of the application 308 may also be implemented by one or more specially designed hardware and firmware components, such as FPGAs, ASICs and the like in other embodiments.

[0033] The memory 304 may also store a repository 312 containing, for example, a map of the environment in which the apparatus 103 operates, for use during the execution of the application 308. The apparatus 103 also includes a communications interface 316 enabling the apparatus 103 to communicate with the server 101 (e.g. via the link 105 or via the dock 106 and the link 107), for example to receive instructions to navigate to specified locations and initiate data capture operations.

[0034] FIG. 3 also illustrates example components of the application 308. As will be apparent to those skilled in the art, the illustrated components may be implemented as a suite of distinct applications in other embodiments. In the present example, the application 308 includes a task manager 320 configured to receive the instructions from the server 101 defining tasks to be performed by the apparatus 103. The instructions include task fragments, each identifying a task (e.g. a data capture task) and a set of sub-regions in the facility. In the present example, the sub-regions correspond to shelf modules 110. Further, the task fragments typically identify contiguous sets of modules 110. The task manager 320, in general, controls the other components of the application 308 to execute the task fragments according to a sequence that may be updated dynamically under certain conditions.

[0035] The other components of the application 308 include a navigator 324 that determines an initial sequence for the above-mentioned task fragments (e.g. for returning to the task manager 320). The navigator 324 also receives instructions from the task manager 320 to execute task fragments, and generates paths (for example, based on the map stored in the repository 312) through the retail facility traversing the

shelf modules 110 identified by each task fragment. The navigator 324 further controls the locomotive mechanism 203 to travel along the above-mentioned paths.

[0036] The application 308 includes a data capture controller 328 that receives instructions from the task manager 320 to execute task fragments, and controls the sensors 108 (e.g. the image sensors 207 and the depth sensors 209) to capture data such as images of the shelf modules 110 while the apparatus 103 travels along the paths under the action of the navigator 324.

[0037] The above-mentioned paths include taxi portions, which are portions of the paths that the apparatus 103 travels along without performing tasks (e.g. data capture), and execution portions, which are portions of the paths that traverse modules 110 where the specified task is to be performed. For example, for a data capture task relating to a particular region (also referred to as an aisle, e.g. a set of contiguous shelf modules 110), the apparatus 103 may travel along a taxi portion of a path to arrive at an end of the aisle, and then along an execution portion that traverses the aisle while performing the data capture task. The dynamic updating of the sequence of task fragments noted above is performed by the task manager 320 to reduce the length of the taxi portions (i.e. to reduce the time spent traveling by the apparatus 103 during which the specified task is not being performed).

[0038] The functionality of the application 308 to dynamically update the sequence in which task fragments are executed by the apparatus 103 will now be described in greater detail, with reference to FIG. 4. FIG. 4 illustrates a method 400 of dynamic task sequencing, which will be described in conjunction with its performance in the system 100, and in particular by the apparatus 103, with reference to the components illustrated in FIGS. 2 and 3. As will be apparent in the discussion below, in other examples, some or all of the processing performed by the server 101 may be performed by the apparatus 103, and some or all of the processing performed by the apparatus 103 may be performed by the server 101.

[0039] At block 405, the apparatus 103, and in particular the task manager 320, receives a plurality of task fragments and a task identifier. The task fragments and the task identifier may be received, for example, from the server 101 via the communications interface 316. The task fragments received at block 405 each include a set of sub-region

identifiers. More specifically, in the present example each task fragment identifies a set of contiguous shelf modules 110, and the task identifier indicates a task to be performed by the apparatus 103 with respect to the identified shelf modules 110.

[0040] FIG. 5 illustrates an example retail facility including a plurality of shelf modules 110, referred to by alphanumeric identifiers for ease of reference in the discussion below. In particular, six sets of contiguous modules 110 are shown, namely A1-A5, B1-B5, C1-C5, D1-D5, E1-E5 and F1-F5. A current position of the apparatus 103 is also shown, and a current heading direction 500 the apparatus 103 is also illustrated. As will be apparent to those skilled in the art, the location and heading 500 of the apparatus 103 constitute the pose of the apparatus 103. Also shown in FIG. 5 is an obstacle 502 at the end of the aisle containing the "B" and "C" modules. The obstacle 502 can be a dynamic obstacle not represented in the map stored in the repository 312, such as a person, or a fixed obstacle such as a pillar (which may be represented in the map stored in the repository 312).

[0041] Table 1 contains example task data received at the apparatus 103 from the server 101 in an example performance of block 405, defining task fragments to be performed by the apparatus 103 in the retail facility shown in FIG. 5. Although the task fragments are shown in a tabular format for ease of illustration, it will be apparent to those skilled in the art that the task fragments and the task identifier can be provided to the apparatus in any of a wide variety of suitable formats.

[0042] Table 1 contains a task identifier, indicating that the apparatus 103 is to perform a data capture task with respect to each of the task fragments. In other examples, a plurality of task identifiers may be provided to the apparatus 103, for example specifying different tasks for different task fragments. Table 1 also contains task fragments identifying certain sets of the modules 110 shown in FIG. 5. Specifically, Table 1 lists five task fragments, corresponding respectively to the "A", "B", "C", "D" and "E" modules shown in FIG. 5 (Table 1 does not contain a task fragment corresponding to the "F" modules of FIG. 5).

Table 1: Example Task Fragments

Task ID	Fragment ID	Module ID(s)
Data Capture	A	A1, A2, A3, A4, A5
	B	B1, B2, B3, B4, B5
	C	C1, C2, C3, C4, C5
	D	D1, D2, D3, D4, D5
	E	E1, E2, E3, E4, E5

[0043] As seen above, each task fragment can include a fragment identifier (which may be omitted in other embodiments), and identifiers of each module to which the fragment relates. The module identifiers enable the apparatus 103 (e.g. the navigator 324) to retrieve locations for the corresponding modules from a map in the repository 312.

[0044] Responsive to receiving the task fragments, the task manager 320 stores the task fragments in the memory 304, and also generates execution criteria corresponding to each module identified in the task fragments. As will be discussed in greater detail below, while the apparatus 103 is taxiing, the task manager monitors the current pose (i.e. location and direction 500) of the apparatus 103. When the current pose of the apparatus 103 satisfies the execution criteria corresponding to a given module, the task manager 320 updates the sequence of task fragments to initiate the performance of the specified task for that module. In other words, the task manager 320 interrupts taxiing of the apparatus 103 to begin task execution when the current pose of the apparatus 103 satisfies any of the execution criteria.

[0045] The execution criteria for a given module therefore represent physical constraints that are satisfied when the apparatus 103 is in a position favorable to the performance of the task for that module. In the present example, the execution criteria for each module define a proximity criterion and a direction criterion. The proximity criterion defines an area of the facility adjacent to the relevant module, and the direction criterion defines a direction of travel. When the current pose of the apparatus 103 indicates that the apparatus 103 is inside the area defined by the proximity criterion, and is traveling in a direction matching the direction criterion (e.g. within a certain angular threshold), the execution criteria are satisfied.

[0046] FIG. 5 illustrates execution criteria 504 for the “C” and “D” modules, superimposed on the facility. The task manager 320 generates execution criteria for

every module identified in the task fragments, but criteria 504 are illustrated only for the "C" and "D" modules to preserve legibility, as many of the criteria overlap spatially. Specifically, FIG. 5 illustrates execution criteria 504-C1 through 504-C5 for the "C" modules, and execution criteria 504-D1 through 504-D5 for the "D" modules. As illustrated, each execution criterion defines an area 508 within the facility, and a direction 512. The areas 508 can be defined by coordinates, such as a set of four coordinates in the frame of reference 102. Although the areas 508 are shown as rectangular areas in FIG. 5, the areas 508 can also have various other shapes. As will be apparent from FIG. 5, each area 508 is adjacent to at least a portion of the corresponding module. That is, the are 508 of the criteria 504-C1 is adjacent to the module C1 to which the criteria 504-C1 corresponds.

[0047] The direction 512 of each of the execution criteria 504 indicates a direction of travel required for the apparatus 103 to perform the data capture task with respect to the corresponding module. The mast 205 and the sensors 108 are fixed to the chassis 201 of the apparatus 103, and therefore the apparatus 103, as shown in the illustrated examples, is required to travel with the target module on the right (i.e. starboard) side of the chassis 201. Therefore, the directions 512 of the execution criteria 504-C1 to 504-C5 are opposite to the directions 512 of the execution criteria 504-D1 to 504-D5.

[0048] As will now be apparent, if the apparatus 103 taxis into the region 508 while traveling in the direction 512 (or within a configurable angular threshold of the direction 512) of the execution criteria 504-C1, then the execution criteria 504-C1 are satisfied. The task manager 320, according to a mechanism discussed below, will therefore initiate performance of the data capture task with respect to the module C1, instead of another module towards which the apparatus 103 was taxiing.

[0049] Returning to FIG. 4, having received the task identifier and task fragments, and having generated and stored the execution criteria, the apparatus 103 generates a sequence for the task fragments. In the present example, the navigator 324 retrieves the locations of the sub-regions (i.e. modules 110) identified in the task fragments from the map stored in the repository 312. The navigator 324 also retrieves the current location of the apparatus 103 itself in the common frame of reference 102. Based on the locations of the modules 110 identified by the task fragments (e.g., the locations of the end-points

of each task fragment, such as the location of the modules A1 and A5 for the task fragment A) and the current location of the apparatus 103, the navigator 324 generates the sequence to optimize the distance that the apparatus 103 must travel to complete the task fragments. That is, the navigator 324 sequences the task fragments to reduce the total distance travelled in the execution of the task fragments, according to a suitable path optimization process.

[0050] Responsive to generating the sequence, the navigator 324 returns the sequence to the task manager 320. Table 2 contains an example sequence returned to the task manager 320 by the navigator 324.

Table 2: Example Task Fragment Sequence

Fragment ID	Module ID	Status
A	A1	Active
	A2	
	A3	
	A4	
	A5	
B	B5	
	B4	
	B3	
	B2	
	B1	
C	C1	
	C2	
	C3	
	C4	
	C5	
D	D5	
	D4	
	D3	
	D2	
	D1	
E	E1	
	E2	
	E3	
	E4	
	E5	

[0051] As seen above, the sequence generated by the navigator 324 defines an order in which the apparatus 103 will perform the task fragments shown in Table 1. The sequence may specify, as in the illustrated embodiment, not only the order of the task

fragments, but also the order of the modules identified by each task fragment. Thus, Table 2 defines a sequence in which the apparatus 103 will first traverse the "A" modules beginning at the module A1 and ending at the module A5. The apparatus 103 will then traverse the "B" modules beginning at the module B5 and ending at the module B1, and so on until the module E5 is reached. As will be apparent from FIG. 5, the order specified for the task fragments (and the modules identified therein) minimizes taxiing between task fragments.

[0052] At block 410, the task manager 320 selects an active task fragment according to the sequence. The active task fragment selected is the task fragment nearest the beginning of the sequence that has not yet been completed. Therefore, in the present example, the task fragment A is selected at block 410. The task manager 320 can store an indication of which task fragment is active, as shown in Table 2. The task manager 320 can also, in some examples, store an indication of which module of the active task fragment is currently active.

[0053] At block 415, the apparatus 103 generates a path for the active fragment. For example, the task manager 320 can instruct the navigator 324 to generate a path that traverses the modules of the active task fragment (i.e. the modules A1 to A5, in the present example). Turning to FIG. 6, the modules 110 are shown along with the apparatus 103 and a path 600 generated by the navigator 324 at block 415. The generation of a path at block 415 can include generating a path to a starting location for the task fragment (e.g. from the current location of the apparatus 103 to the module A1, in the present example), in addition to generating a path traversing the sub-regions of the task fragment itself. The path 600 therefore includes a taxi portion 604 from the current location of the apparatus 103, and an execution portion 608 that traverses the modules A1 to A5, for performance of the data capture task. In some embodiments, the portions 604 and 608 are generated as distinct paths (e.g. responsive to separate instructions from the task manager 320 to the navigator 324).

[0054] Referring again to FIG. 4, following path generation, at block 420 the task manager 320 determines whether a proportion of the modules of the active task fragment that can only be reached from the current location of the apparatus 103 by taxiing (e.g. by taxiing any distance, or in some embodiments by taxiing more than a

configurable threshold distance) exceeds a configurable threshold. In other words, for the task fragment A, the task manager 320 determines a proportion of the modules A1-A5 that can only be reached via the taxi portion 604 of the path 600. Those modules are referred to as "inaccessible" modules, although as will be understood, they are simply inaccessible without taxiing, rather than being entirely inaccessible. That proportion is compared to a threshold (e.g. 20%, although a wide variety of other thresholds are also contemplated). As seen in FIG. 6, all of the modules A1-A5 can only be reached via the taxi portion 604 of the path (i.e. the proportion of inaccessible modules is 100%), and the determination at block 420 is therefore negative. When the determination is affirmative, the task manager 320 updates the status of the inaccessible modules at block 425, as will be discussed further below.

[0055] In the present example, following a negative determination at block 420, the apparatus 103 proceeds to block 430. At block 430 the task manager 320 initiates the execution of the active task fragment selected at block 410 by sending instructions to the navigator 324 and the data capture controller 328 to travel along the path 600 generated at block 415 and to begin capturing data, respectively. More specifically, the task manager 320 is configured to transmit an instruction to begin navigation for the active task fragment to the navigator 324, and to transmit an instruction to begin data capture for the active task fragment to the data capture controller 328.

[0056] The navigator 324 is configured, during the performance of block 430, to control the locomotive mechanism 203 and associated components of the apparatus 103, to travel along the path 600 generated at block 415 for the active task fragment. The navigator 324 achieves such control via one or more navigation sensors (e.g., lidar sensors), the map stored in the repository 312 and the current pose of the apparatus 103, which is periodically updated. During the traverse of the modules 110 corresponding to the active task fragment, the data capture controller 328 is configured to control any one or more of the sensors 108 (e.g. the cameras 207 and depth sensors 209) to capture image data depicting the sub-regions the apparatus 103 is traversing and store the image data in the repository 312.

[0057] As the apparatus 103 travels along the sub-regions corresponding to the active task fragment, the navigator 324 is configured to monitor the current pose of the

apparatus 103. The current pose is periodically reported to the task manager 320. The navigator 324 also notifies the task manager 320 of each sub-region that has been successfully traversed. Thus, the task manager 320 can update status data corresponding to the modules of the active task fragment.

[0058] During travel according to the path 600, at block 435 the task manager 320 determines, based on the current pose of the apparatus 103 received from the navigator 324, whether any of the execution criteria generated at block 405 are satisfied. That is, the task manager 320 compares the current location and direction of the apparatus 103 with the areas 508 and directions 512. In particular, the comparison of current pose to execution criteria 504 at block 435 is performed only during taxiing (i.e. while traveling the taxi portion 604 of the path 600). Further, the execution criteria 504 corresponding to the active task fragment are omitted from the comparison at block 435. Thus, while traveling the taxi portion 604, the task manager 320 does not assess whether the execution criteria for any of the modules A1-A5 are satisfied.

[0059] When the determination at block 435 is affirmative, the task manager 320 updates the task fragment sequence at block 440, as will be discussed further below. In the present example, as will be apparent from FIG. 6, the determination at block 435 is negative as the apparatus 103 travels the taxi portion 604, as no execution criteria 504 for non-active modules will be satisfied. The apparatus 103 therefore proceeds to block 445.

[0060] At block 445 the navigator 324 determines whether the path 600 is obstructed. When an obstacle is detected, the navigator 324 generates an updated path at block 415 (e.g. for the modules 110 remaining to be traversed). When no obstacle is detected, execution of the path 600 continues. At block 450, when the active task fragment is complete (i.e. when each module 110 of the active task fragment has been traversed or marked incomplete, as will be discussed below), the task manager 320 selects the next task fragment from the sequence. Thus, referring to FIG. 7, in which the apparatus 103 is shown having reached the end of the path 600, the task manager 320 selects the next task fragment from the sequence of Table 2, and performance of the method 400 returns to block 415. Table 3 contains status data indicating that the modules A1-A5 are complete (and the modules A1-A5 are illustrated with shading to indicate completion

in FIG. 7). Further, Table 3 indicates that the task fragment B has been selected as the active task fragment.

Table 3: Updated Task Fragment Sequence

Fragment ID	Module ID	Status
A	A1	Complete: A1-A5
	A2	
	A3	
	A4	
	A5	
B	B5	Active
	B4	
	B3	
	B2	
	B1	
C	C1	
	C2	
	C3	
	C4	
	C5	
D	D5	
	D4	
	D3	
	D2	
	D1	
E	E1	
	E2	
	E3	
	E4	
	E5	

[0061] At block 415, the navigator 324 generates a path for the active task fragment, shown as a path 700 in FIG. 7. Of particular note, the obstacle 502 prevents the apparatus 103 from reaching the module B5 without taxiing back towards the module A1 before returning along the next aisle to reach the module B5. Thus, the path 700 includes a lengthy taxi portion 704, followed by an execution portion 708. The determination at block 420 is negative, as none of the "B" modules can be reached without taxiing. Thus at block 430 the apparatus 103 begins traveling along the path 700.

[0062] Referring now to FIG. 8, when the apparatus 103 has traveled along part of the taxi portion 704, the current pose of the apparatus 103 satisfies the execution criteria 504-C1 corresponding to the module C1. The determination at block 435 is therefore affirmative. At block 440 the task manager 320 updates the sequence and selects a new active segment based on the updated sequence. In particular, the task manager 320 updates the sequence to place the task fragment C above the task fragment B, as shown below in Table 4, and to select the task fragment C as the active task fragment. Performance of the method 400 then returns to block 415, at which the task manager 320 instructs the navigator 324 to generate a path for the task fragment C.

Table 4: Updated Task Fragment Sequence

Fragment ID	Module ID	Status
A	A1	Complete: A1-A5
	A2	
	A3	
	A4	
	A5	
C	C1	Active
	C2	
	C3	
	C4	
	C5	
B	B5	
	B4	
	B3	
	B2	
	B1	
D	D5	
	D4	
	D3	
	D2	
	D1	
E	E1	
	E2	
	E3	
	E4	
	E5	

[0063] Turning to FIG. 9, a path 900 is shown, including a taxi portion 904 and an execution portion 908, for completing the task fragment C. At block 430, the apparatus

103 travels along the path 900 to complete the data capture task for each of the modules C1-C5. As will be apparent, the determination at block 435 is negative, as is the determination at block 445. When the path 900 is complete, the task manager 320 selects the next task fragment according to the updated sequence (i.e. the task fragment B in the present example). Repeating blocks 415 to 450 for the task fragment B leads to completion of the task fragment B, and the selection of the next task fragment, which in the present example is the task fragment D.

[0064] FIG. 10 illustrates the apparatus 103 following the completion of the task fragment B and selection of the task fragment D as the active task fragment. Table 5 illustrates the updated sequence and status of the task fragments.

Table 5: Updated Task Fragment Sequence

Fragment ID	Module ID	Status
A	A1	Complete: A1-A5
	A2	
	A3	
	A4	
	A5	
C	C1	Complete: C1-C5
	C2	
	C3	
	C4	
	C5	
B	B5	Complete: B1-B5
	B4	
	B3	
	B2	
	B1	
D	D5	Active
	D4	
	D3	
	D2	
	D1	
E	E1	
	E2	
	E3	
	E4	
	E5	

[0065] FIG. 10 also illustrates a path 1000 generated at block 415 for the task fragment D, including a taxi portion 1004 and an execution portion 1008. As will be apparent from FIG. 10, during travel along the taxi portion 1004 towards the module D5, the apparatus 103 will satisfy the execution criteria 504-E1 corresponding to the module E1. Therefore, following another affirmative determination at block 435, the task manager 320 updates the sequence to place the task fragment E ahead of the task fragment D, and initiates path generation at block 415 for the task fragment E. Table 6 illustrates the further updated sequence

Table 6: Updated Task Fragment Sequence

Fragment ID	Module ID	Status
A	A1	Complete: A1-A5
	A2	
	A3	
	A4	
	A5	
C	C1	Complete: C1-C5
	C2	
	C3	
	C4	
	C5	
B	B5	Complete: B1-B5
	B4	
	B3	
	B2	
	B1	
E	E1	Active
	E2	
	E3	
	E4	
	E5	
D	D5	
	D4	
	D3	
	D2	
	D1	

[0066] Turning to FIG. 11, a path 1100 is shown as generated at block 415 for the task fragment E, with the apparatus 103 having traveled along part of the path 1100 and completed the traversal of the modules E1, E2 and E3. FIG. 11 also illustrates that an

obstacle 1102, such as a person, has been detected by the apparatus 103. The module E4 is obstructed by the obstacle 1102 (i.e. data capture of the module E4 cannot be completed). Further, the module E5 cannot be reached by continuing to follow the path 1100. Therefore, the determination at block 445 is affirmative, and the apparatus 103 returns to block 415 to generate an updated path to reach the module E5. As seen in FIG. 11, the module E5 can be reached, for example, by traveling back towards the module E1 and then past the modules F1 to F5.

[0067] At block 420, the module E5 is considered an inaccessible module, because it can only be reached by taxiing more than a threshold distance. In some embodiments, the assessment at block 420 can be performed in conjunction with path generation at block 415, and includes determining whether a path can be planned to reach the relevant module(s) (i.e. the module E5 in the present example) without leaving the current aisle. In either implementation, the module E5 is inaccessible. Further, the module E5 constitutes only one fifth (20%) of the task fragment E. Therefore, the determination at block 420 is affirmative, and at block 425 the task manager 320 ignores the inaccessible modules, for example by storing status data indicating that the inaccessible modules have been skipped, as shown in Table 7 below.

[0068] Following the performance of block 425, the apparatus 103 can return to block 415 to generate a path for the active task fragment that omits the ignored modules from block 425. In the present example, it will be apparent that when the module E5 is ignored, no further modules remain to be completed in the task fragment E, and that further path planning is no longer necessary for the task fragment E. The task manager 320 can therefore also determine, following block 425, whether the active task fragment is complete as a result of the performance of block 425. When the active task fragment is complete, the task manager 320 proceeds to block 450 before returning to block 415. In the present example, therefore the task manager 320 selects the task fragment D as the active task fragment, and returns to block 415. The updated status of the task fragments is shown in Table 7.

Table 7: Updated Task Fragment Sequence

Fragment ID	Module ID	Status
A	A1	Complete: A1-A5
	A2	
	A3	
	A4	
	A5	
C	C1	Complete: C1-C5
	C2	
	C3	
	C4	
	C5	
B	B5	Complete: B1-B5
	B4	
	B3	
	B2	
	B1	
E	E1	Complete: E1-E3 Obstructed: E4 Skipped: E5
	E2	
	E3	
	E4	
	E5	
D	D5	Active
	D4	
	D3	
	D2	
	D1	

[0069] As seen above, the task fragment D is indicated as the active task fragment, and the module E5 is indicated as having been skipped (i.e. ignored), as completing data

capture for the module E5 alone would require the apparatus 103 to taxi a substantial distance.

[0070] Following selection of the task fragment D as the active task fragment, a path 1200 as illustrated in FIG. 12 is generated for the task fragment D. As shown in FIG. 12, the modules D4 and D5 have been marked as obstructed and skipped, respectively. Specifically, the module D4 cannot be reached as a result of the obstacle 1102, and the module D5 has been ignored following a further affirmative determination at block 420. More specifically, planning a path from the location of the apparatus 103 shown in FIG. 12 to the module D5 requires a substantial taxiing distance (e.g. above the threshold mentioned earlier), while the modules D3, D2 and D1 can be reached with minimal taxiing (e.g. without leaving the aisle the apparatus 103 is currently in). The module D5 is therefore ignored at block 425, and block 415 is repeated to generate the path 1200 shown. Traveling along the path 1200 results in the completion of data capture for the modules D3, D2 and D1.

[0071] When the task fragment D is complete, the task fragments received at block 405 will have been completed, with the exception of the modules D4, D5, E4 and E5. The task manager 320 can report the status of the task fragments to the server 101, for example, and the server 101 can generate a further set of task fragments for subsequent execution, incorporating any previously incomplete modules (i.e. the modules D4, D5, E4 and E5).

[0072] Through the generation of execution criteria and the performance of block 435, the apparatus 103 may therefore detect conditions in which time spent traveling can instead be used to perform data capture or other tasks, thus reducing the time spent taxiing by the apparatus 103. Further, the determination at block 420 may further avoid suboptimal taxiing by the apparatus to reach isolated modules, instead deferring the performance of tasks for those modules until a later time (e.g. when the obstacle 1102 may no longer be present).

[0073] In the foregoing specification, specific embodiments have been described. However, one of ordinary skill in the art appreciates that various modifications and changes can be made without departing from the scope of the invention as set forth in the claims below. Accordingly, the specification and figures are to be regarded in an

illustrative rather than a restrictive sense, and all such modifications are intended to be included within the scope of present teachings.

[0074] The benefits, advantages, solutions to problems, and any element(s) that may cause any benefit, advantage, or solution to occur or become more pronounced are not to be construed as a critical, required, or essential features or elements of any or all the claims. The invention is defined solely by the appended claims including any amendments made during the pendency of this application and all equivalents of those claims as issued.

[0075] Moreover in this document, relational terms such as first and second, top and bottom, and the like may be used solely to distinguish one entity or action from another entity or action without necessarily requiring or implying any actual such relationship or order between such entities or actions. The terms "comprises," "comprising," “has”, “having,” “includes”, “including,” “contains”, “containing” or any other variation thereof, are intended to cover a non-exclusive inclusion, such that a process, method, article, or apparatus that comprises, has, includes, contains a list of elements does not include only those elements but may include other elements not expressly listed or inherent to such process, method, article, or apparatus. An element proceeded by “comprises …a”, “has …a”, “includes …a”, “contains …a” does not, without more constraints, preclude the existence of additional identical elements in the process, method, article, or apparatus that comprises, has, includes, contains the element. The terms “a” and “an” are defined as one or more unless explicitly stated otherwise herein. The terms “substantially”, “essentially”, “approximately”, “about” or any other version thereof, are defined as being close to as understood by one of ordinary skill in the art, and in one non-limiting embodiment the term is defined to be within 10%, in another embodiment within 5%, in another embodiment within 1% and in another embodiment within 0.5%. The term “coupled” as used herein is defined as connected, although not necessarily directly and not necessarily mechanically. A device or structure that is “configured” in a certain way is configured in at least that way, but may also be configured in ways that are not listed.

[0076] It will be appreciated that some embodiments may be comprised of one or more specialized processors (or “processing devices”) such as microprocessors, digital signal

processors, customized processors and field programmable gate arrays (FPGAs) and unique stored program instructions (including both software and firmware) that control the one or more processors to implement, in conjunction with certain non-processor circuits, some, most, or all of the functions of the method and/or apparatus described herein. Alternatively, some or all functions could be implemented by a state machine that has no stored program instructions, or in one or more application specific integrated circuits (ASICs), in which each function or some combinations of certain of the functions are implemented as custom logic. Of course, a combination of the two approaches could be used.

[0077] Moreover, an embodiment can be implemented as a computer-readable storage medium having computer readable code stored thereon for programming a computer (e.g., comprising a processor) to perform a method as described and claimed herein. Examples of such computer-readable storage mediums include, but are not limited to, a hard disk, a CD-ROM, an optical storage device, a magnetic storage device, a ROM (Read Only Memory), a PROM (Programmable Read Only Memory), an EPROM (Erasable Programmable Read Only Memory), an EEPROM (Electrically Erasable Programmable Read Only Memory) and a Flash memory. Further, it is expected that one of ordinary skill, notwithstanding possibly significant effort and many design choices motivated by, for example, available time, current technology, and economic considerations, when guided by the concepts and principles disclosed herein will be readily capable of generating such software instructions and programs and ICs with minimal experimentation.

[0078] The Abstract of the Disclosure is provided to allow the reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In addition, in the foregoing Detailed Description, it can be seen that various features are grouped together in various embodiments for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus the following claims are hereby

incorporated into the Detailed Description, with each claim standing on its own as a separately claimed subject matter.

Claims

1. A method in a navigational controller, the method comprising:

obtaining (i) a plurality of task fragments identifying respective sets of sub-regions in a facility, and (ii) an identifier of a task to be performed by a mobile automation apparatus at each of the sets of sub-regions;

selecting an active one of the task fragments according to a sequence specifying an order of execution of the task fragments;

generating a path including (i) a taxi portion from a current position of the mobile automation apparatus to the sub-regions identified by the active task fragment, and (ii) an execution portion traversing the sub-regions identified by the active task fragment;

during travel along the taxi portion, determining, based on a current pose of the mobile automation apparatus, whether to initiate execution of another task fragment; and

when the determination is affirmative, updating the sequence to mark the other task fragment as the active task fragment.

2. The method of claim 1, further comprising:

while traveling along the execution portion of the path, performing the identified task.

3. The method of claim 2, wherein the identified task is a data capture task.

4. The method of claim 1, further comprising:

responsive to updating the sequence, repeating generating the path and determining whether to initiate execution of another task fragment.

5. The method of claim 1, further comprising:

responsive to generating the path, identifying inaccessible ones of the set of sub-regions identified by the active task fragment;

determining a proportion of the set of sub-regions represented by the inaccessible sub-regions; and

when the proportion is below a threshold, ignoring the inaccessible sub-regions.

6. The method of claim 5, wherein identifying inaccessible ones of the set of sub-regions includes determining that an inaccessible one of the set of sub-regions cannot be reached without taxiing.

7. The method of claim 1, further comprising:

for each sub-region identified by each task fragment, generating execution criteria;

wherein determining whether to initiate execution of another task fragment includes comparing the current pose of the mobile automation apparatus to the execution criteria.

8. The method of claim 7, wherein the execution criteria include a proximity criterion and a directional criterion.

9. The method of claim 1, wherein obtaining the task fragments includes receiving the task fragments at the mobile automation apparatus from a server.

10. A mobile automation apparatus, comprising:

a locomotive assembly;

a data capture sensor; and

a navigational controller configured to:

obtain (i) a plurality of task fragments identifying respective sets of sub-regions in a facility, and (ii) an identifier of a task to be performed at each of the sets of sub-regions;

select an active one of the task fragments according to a sequence specifying an order of execution of the task fragments;

generate a path including (i) a taxi portion from a current position of the mobile automation apparatus to the sub-regions identified by the active task fragment, and (ii) an execution portion traversing the sub-regions identified by the active task fragment;

control the locomotive assembly to travel along the path;

during travel along the taxi portion, determine, based on a current pose of the mobile automation apparatus, whether to initiate execution of another task fragment; and

when the determination is affirmative, update the sequence to mark the other task fragment as the active task fragment.

11. The mobile automation apparatus of claim 10, wherein the navigational controller is further configured, during travel along the execution portion of the path, perform the identified task.

12. The mobile automation apparatus of claim 11, wherein the identified task is a data capture task, and wherein the navigational controller is configured to control the data capture sensor to perform the data capture task.

13. The mobile automation apparatus of claim 10, wherein the navigational controller is further configured, responsive to updating of the sequence, to repeat the generation of the path and the determination of whether to initiate execution of another task fragment.

14. The mobile automation apparatus of claim 10, wherein the navigational controller is further configured, responsive to generation of the path, to:

identify inaccessible ones of the set of sub-regions identified by the active task fragment;

determine a proportion of the set of sub-regions represented by the inaccessible sub-regions; and

when the proportion is below a threshold, ignore the inaccessible sub-regions.

15. The mobile automation apparatus of claim 14, wherein the navigational controller is further configured, in order to identify the inaccessible ones of the set of sub-regions, to determine that an inaccessible one of the set of sub-regions cannot be reached without taxiing.

16. The mobile automation apparatus of claim 10, wherein the navigational controller is further configured, for each sub-region identified by each task fragment, to generate execution criteria; and

wherein the navigational controller is further configured, in order to determine whether to initiate execution of another task fragment, to compare the current pose of the mobile automation apparatus to the execution criteria.

17. The mobile automation apparatus of claim 16, wherein the execution criteria include a proximity criterion and a directional criterion.

18. The mobile automation apparatus of claim 10, wherein the navigational controller is configured, in order to obtain the task fragments, to receive the task fragments from a server.

19. A method in a navigational controller, the method comprising:

obtaining a sequence of task fragments identifying respective sets of sub-regions in a facility;

selecting an active one of the task fragments according to the sequence;

controlling a mobile automation apparatus to perform a task at the set of sub-regions identified by the active task fragment;

responsive to a change in a pose of the mobile automation apparatus during performance of the task, updating the sequence to mark another task fragment as the active task fragment.

20. The method of claim 19, further comprising:

responsive to detection of an obstacle by the mobile automation apparatus during performance of the task, initiating the change in the pose to avoid the obstacle.

21. The method of claim 19 wherein performance of the task further comprises controlling the mobile automation apparatus to navigate to the set of sub-regions in the facility.

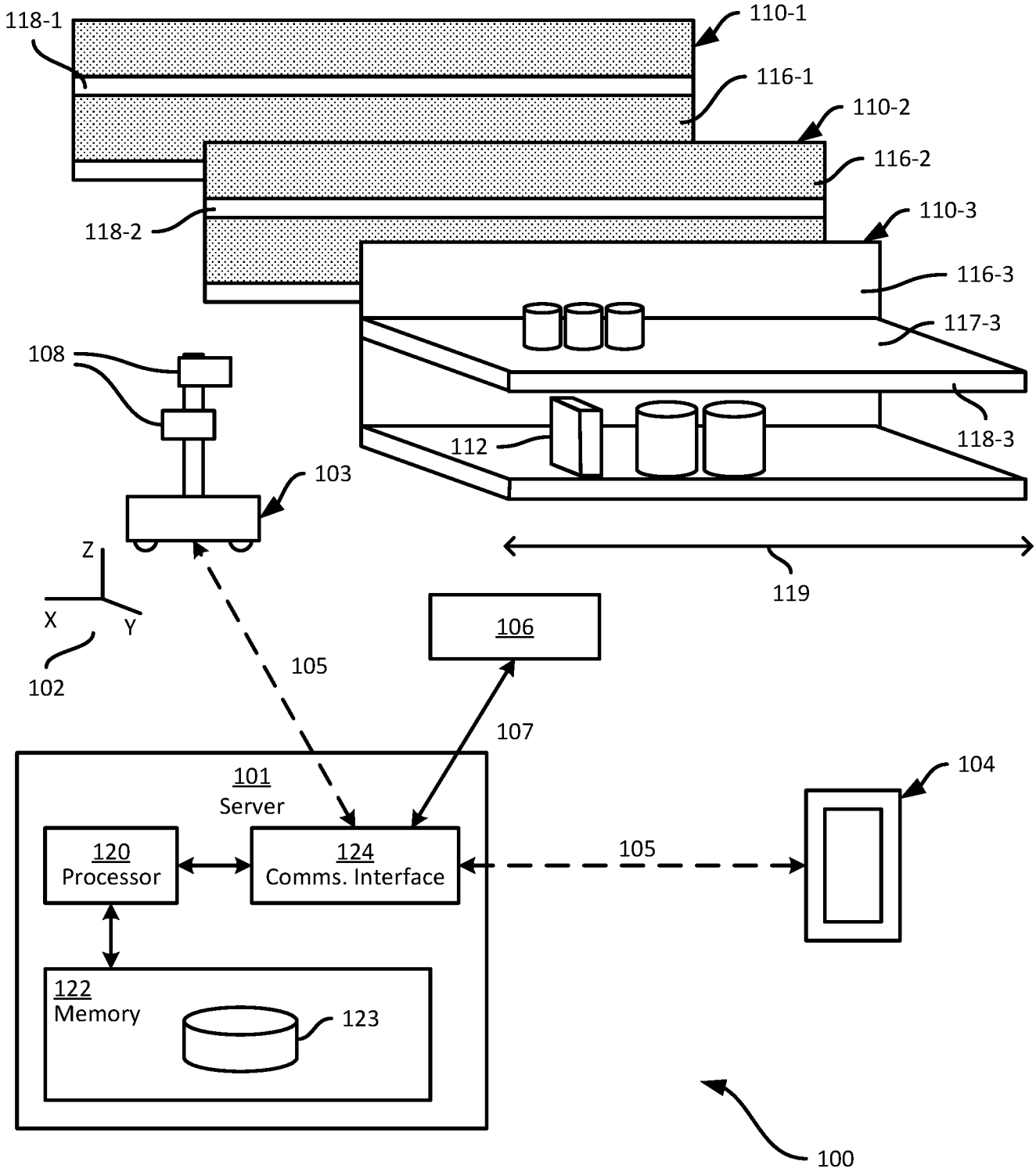

118-1
110-1
116-1
110-2
116-2
118-2
110-3
116-3
117-3
108
118-3
112
103
Z
X
Y
102
119
106
105
107
101
Server
104
120
Processor
124
Comms. Interface
105
122
Memory
123
100

FIG. 1

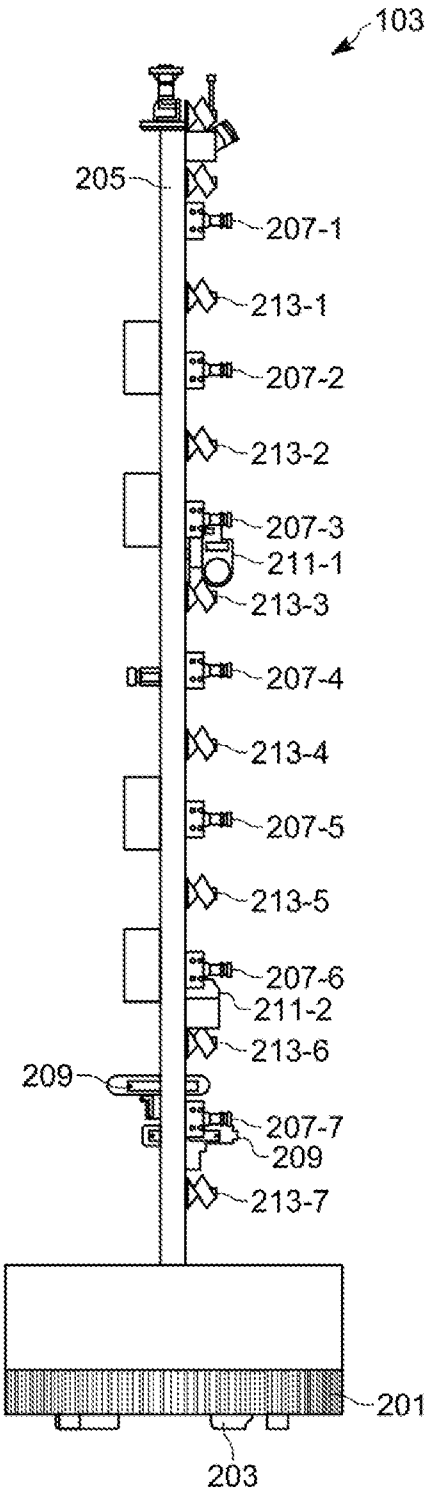

103
205
207-1
213-1
207-2
213-2
207-3
211-1
213-3
207-4
213-4
207-5
213-5
207-6
211-2
213-6
209
207-7
209
213-7
201
203

FIG. 2

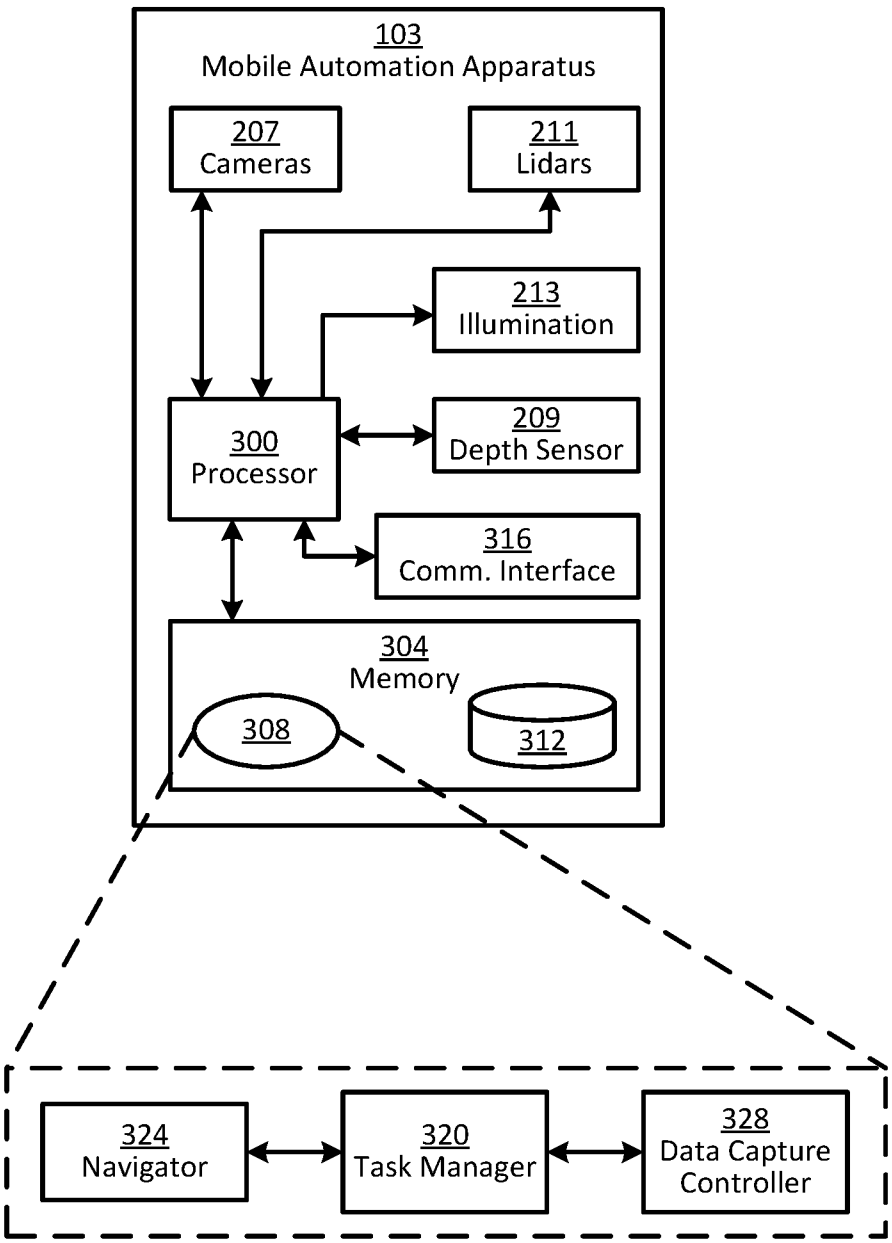

103
Mobile Automation Apparatus
207
Cameras
211
Lidars
213
Illumination
300
Processor
209
Depth Sensor
316
Comm. Interface
304
Memory
308
312
324
Navigator
320
Task Manager
328
Data Capture
Controller

FIG. 3

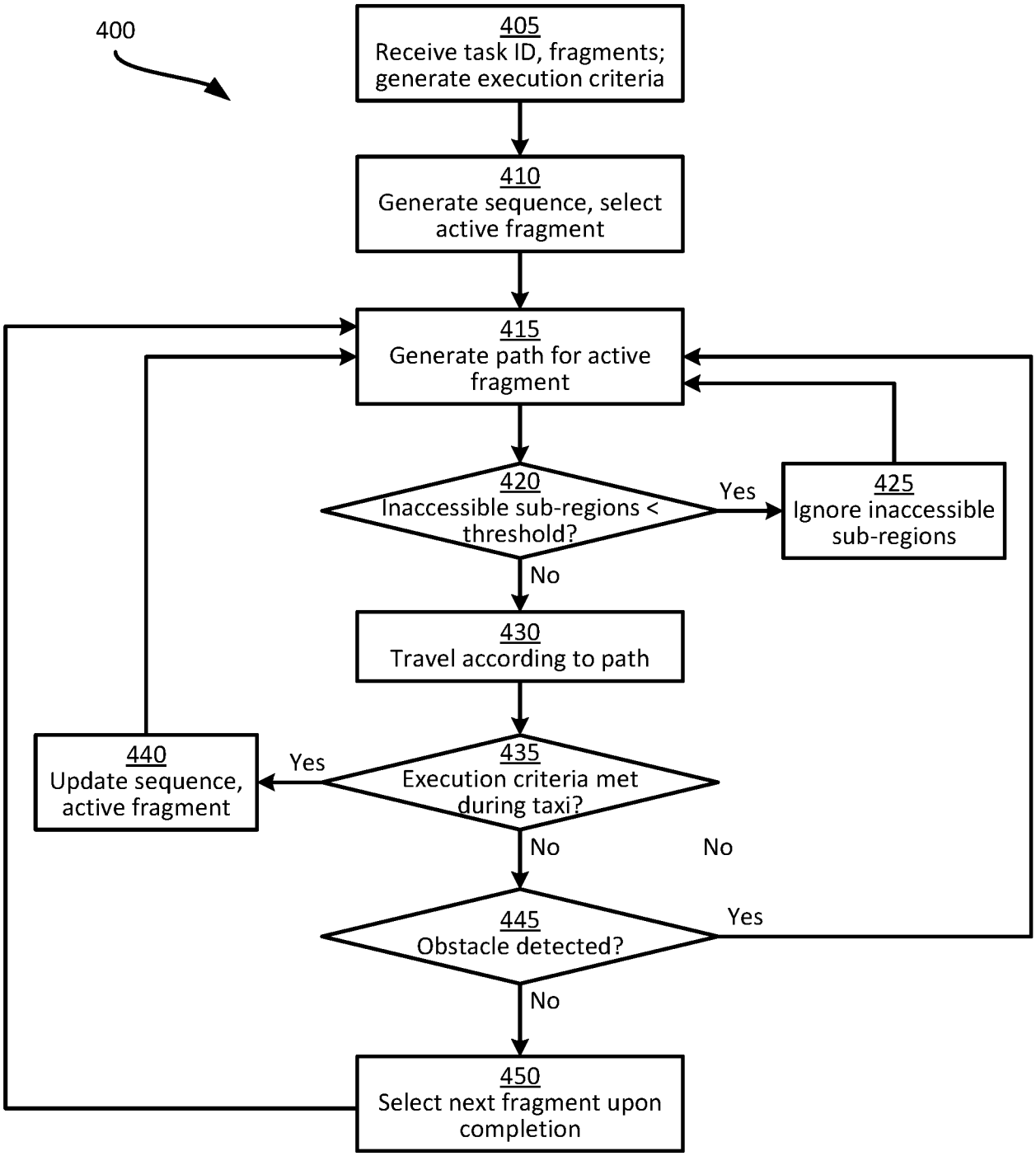

400
405
Receive task ID, fragments; generate execution criteria
410
Generate sequence, select active fragment
415
Generate path for active fragment
420
Inaccessible sub-regions < threshold?
Yes
425
Ignore inaccessible sub-regions
No
430
Travel according to path
435
Execution criteria met during taxi?
Yes
440
Update sequence, active fragment
No
No
445
Obstacle detected?
Yes
No
450
Select next fragment upon completion

FIG. 4

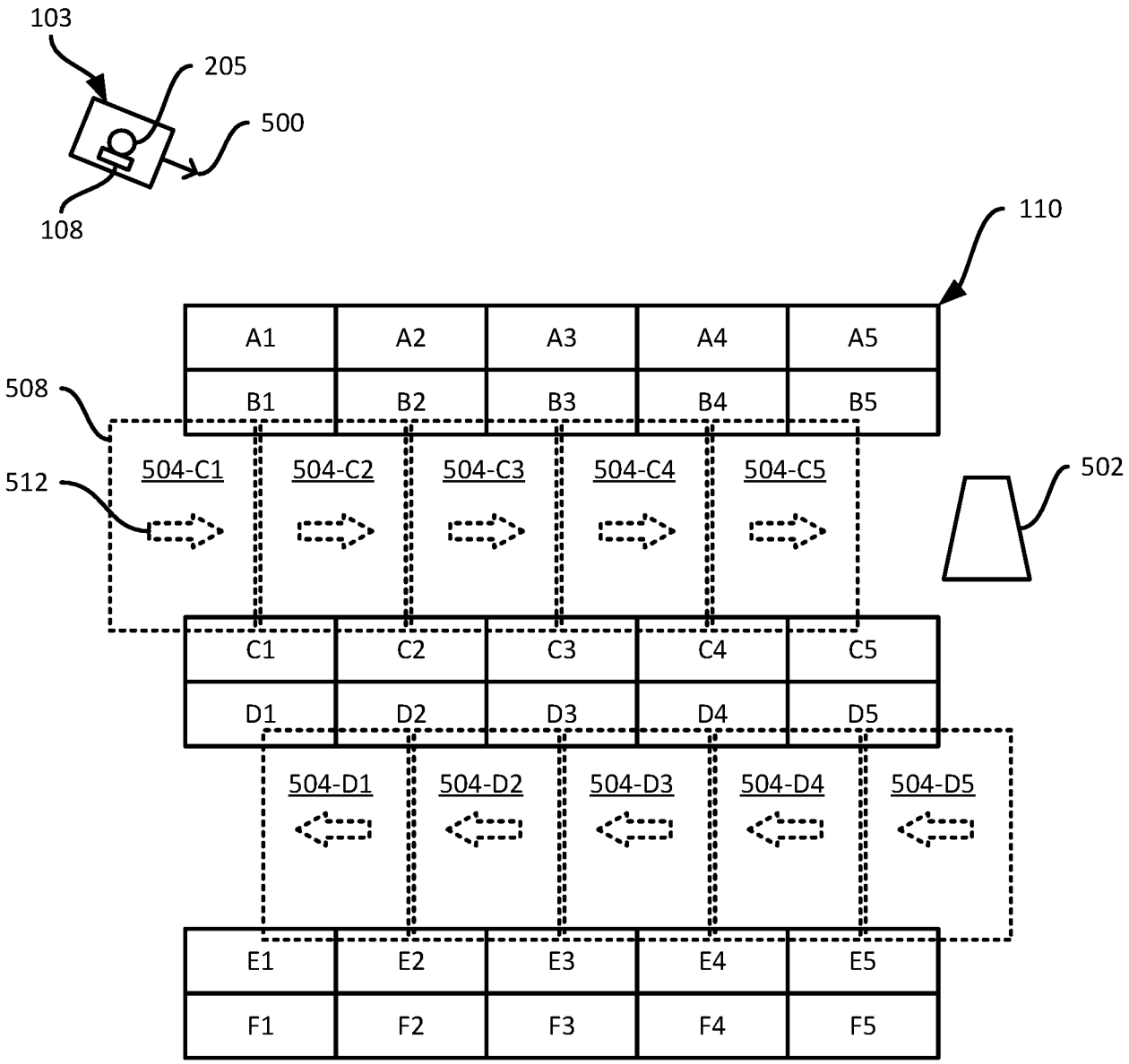

103
205
500
108
110
A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
508
512
504-C1
504-C2
504-C3
504-C4
504-C5
502
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
504-D1
504-D2
504-D3
504-D4
504-D5
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5

FIG. 5

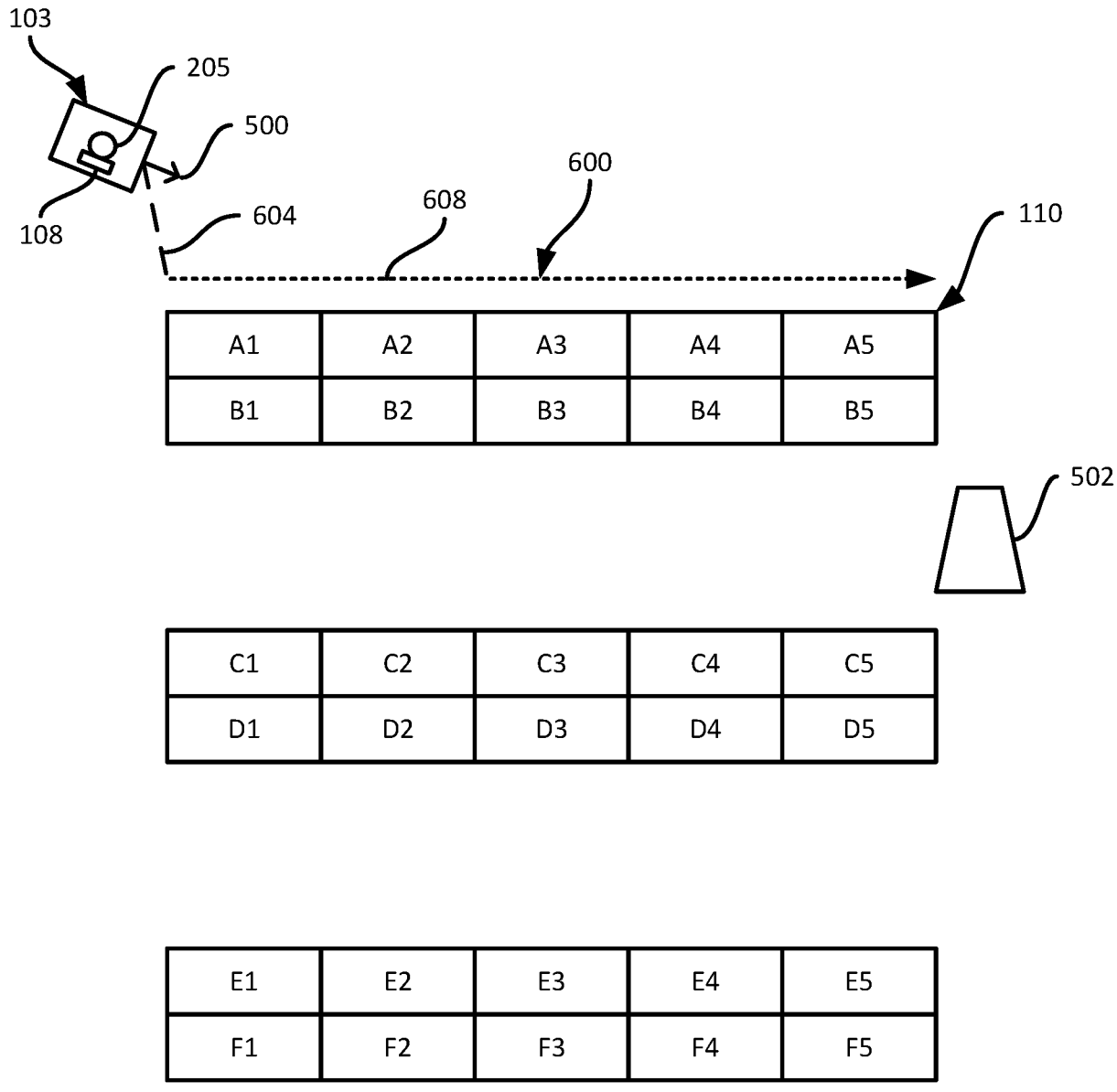

103
205
500
600
608
604
108
110
A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
502
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5

FIG. 6

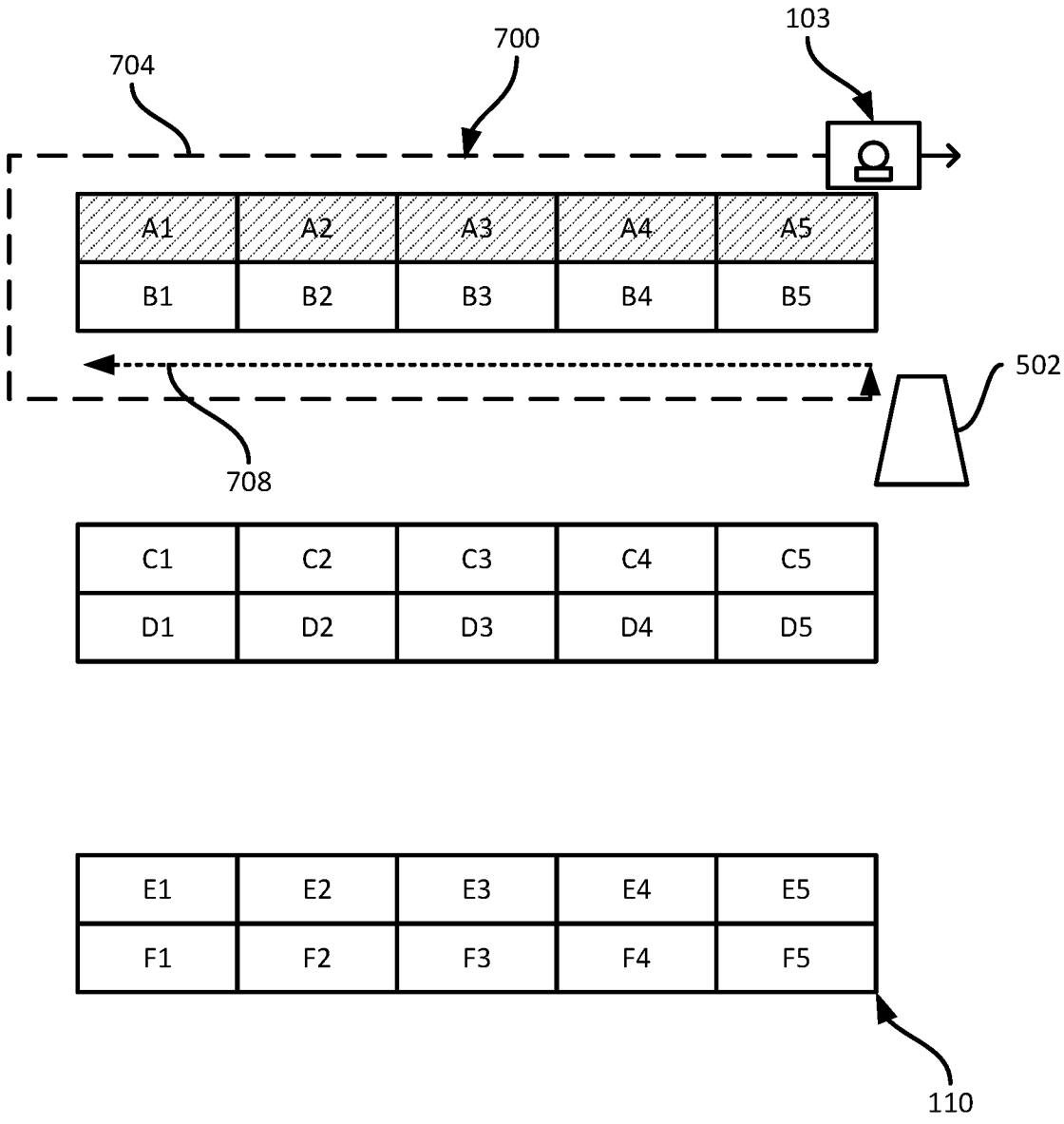

704
700
103
A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
502
708
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5
110

FIG. 7

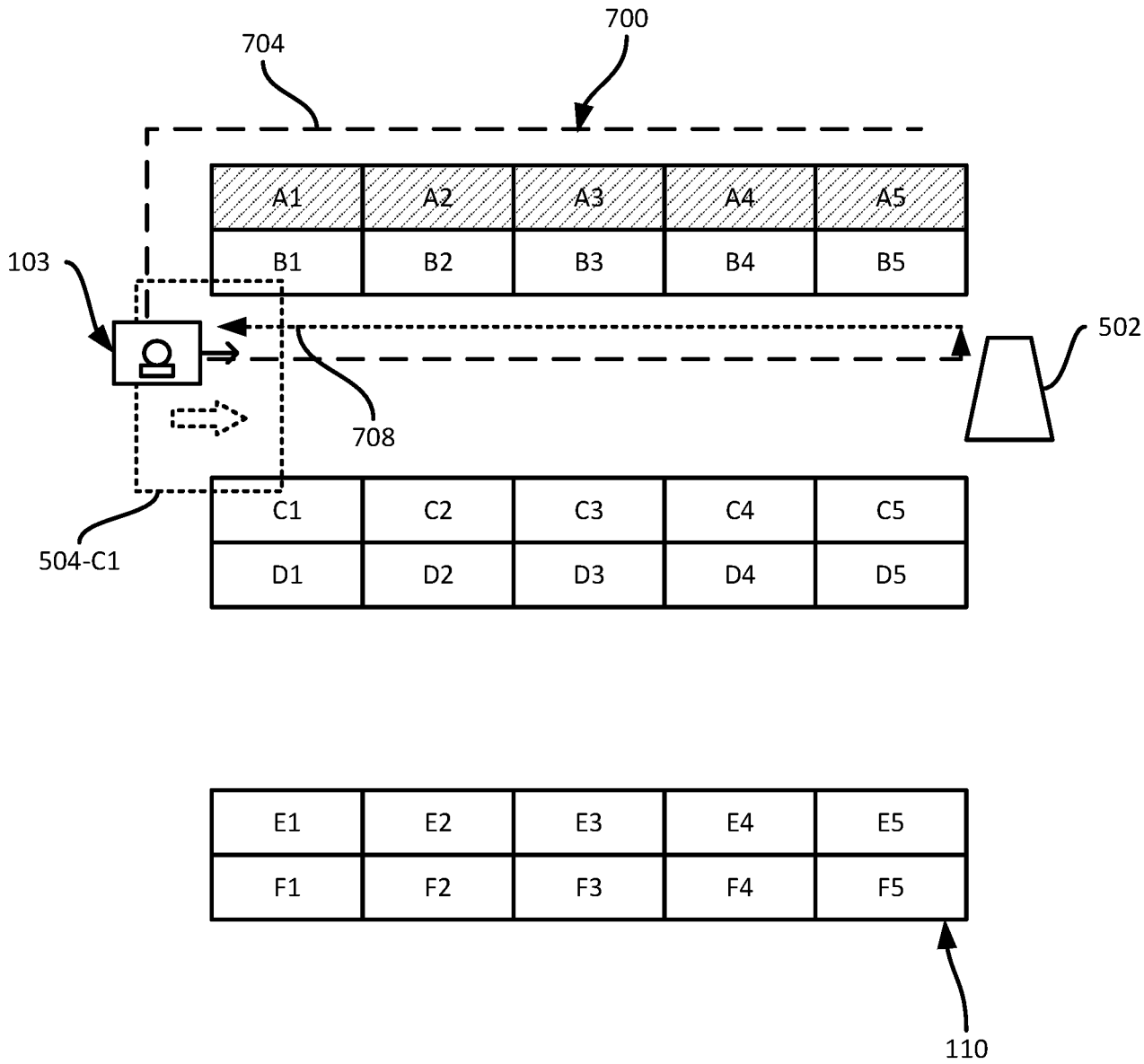

700
704
A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
103
502
708
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
504-C1
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5
110

FIG. 8

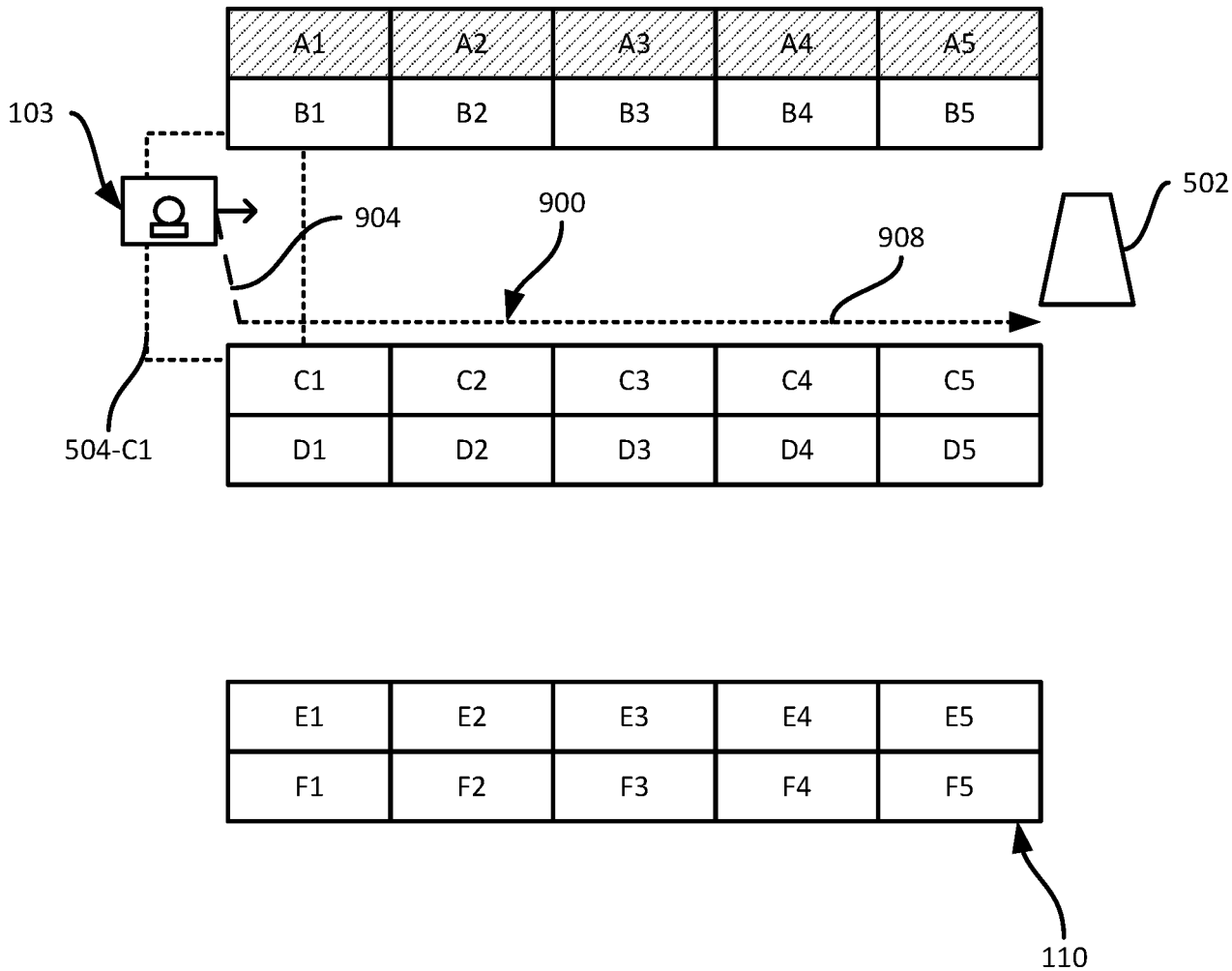

A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
103
904
900
908
502
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
504-C1
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5
110

FIG. 9

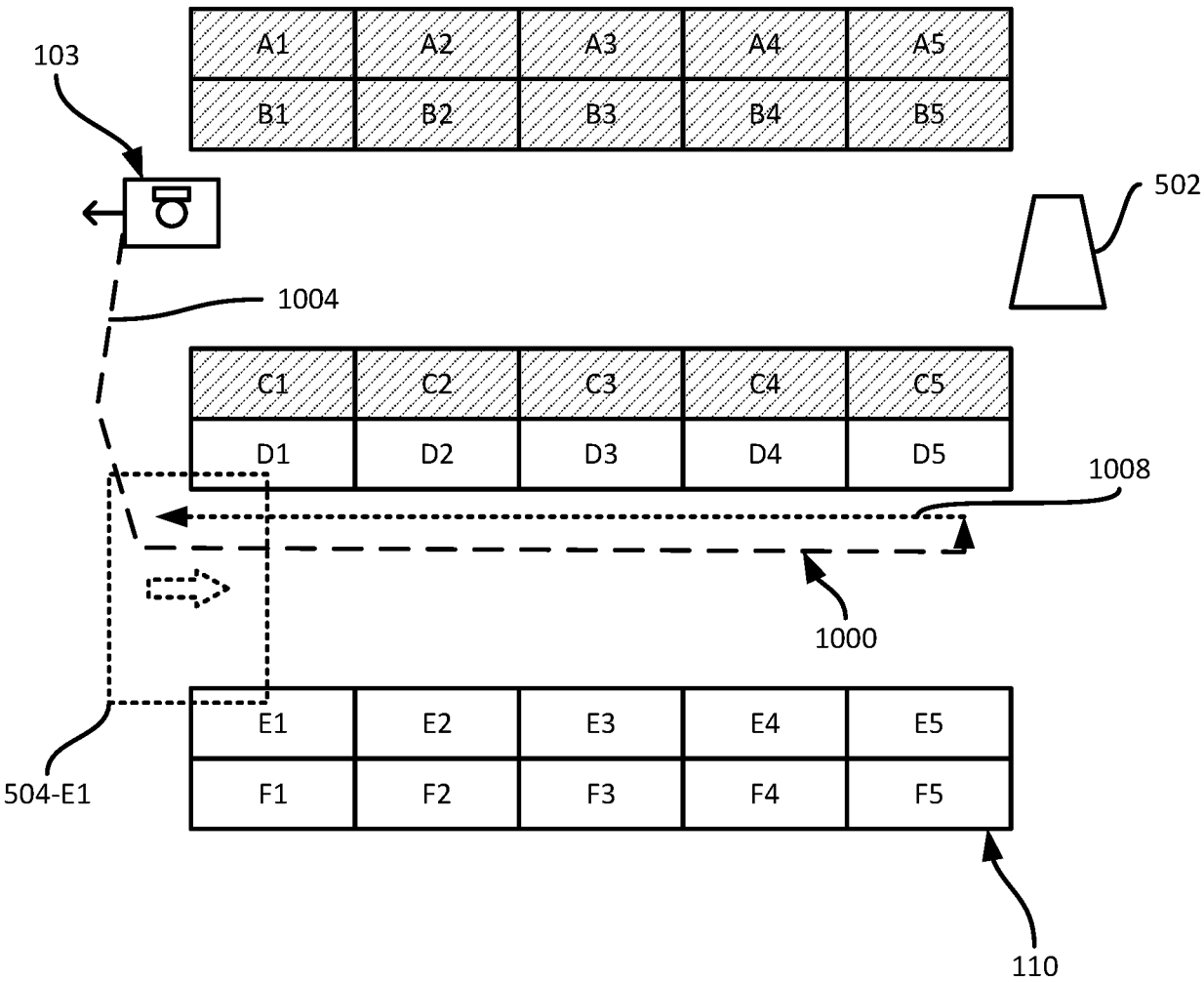

A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
103
502
1004
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
1008
1000
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5
504-E1
110

FIG. 10

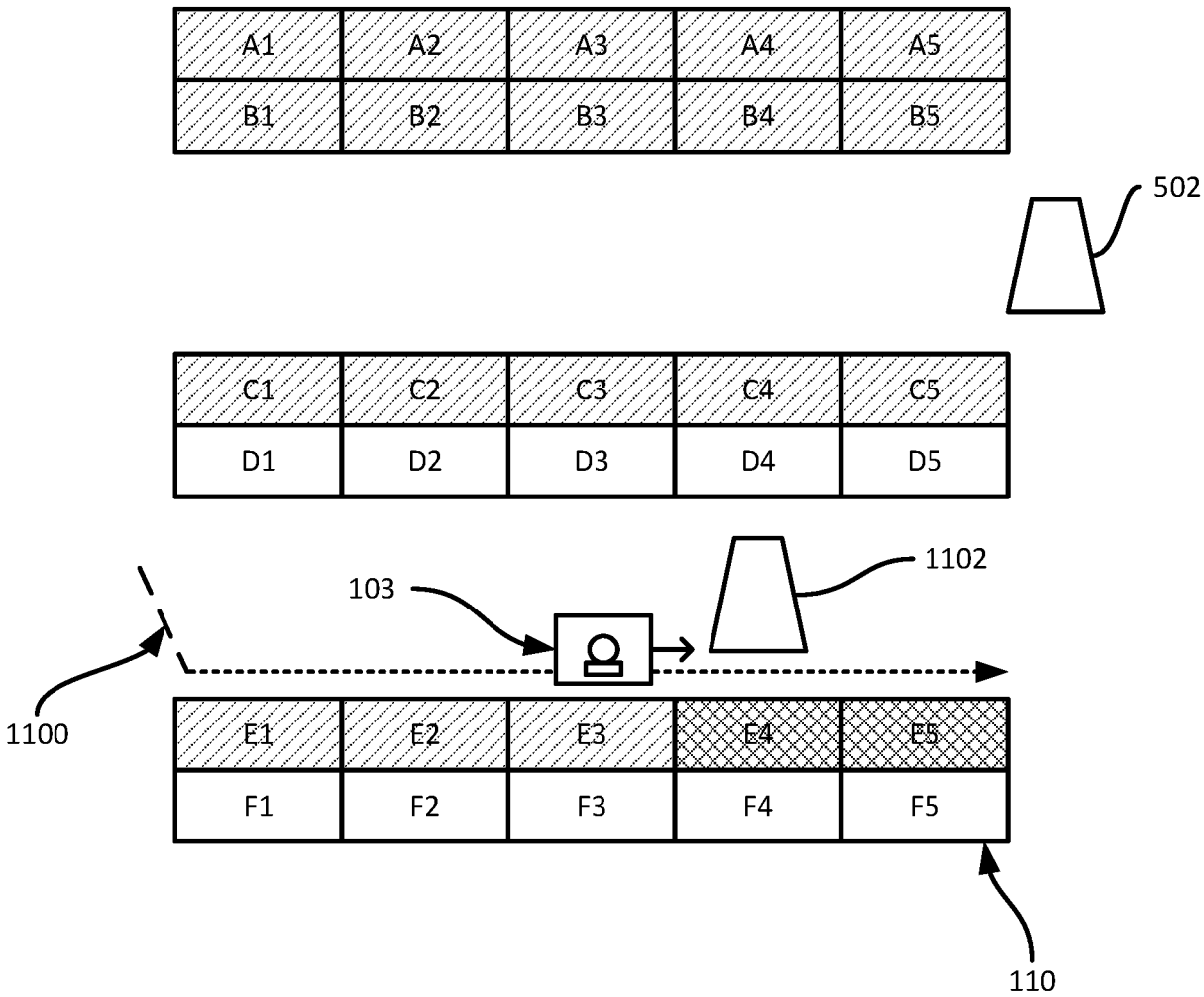

A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
502
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
103
1102
1100
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5
110

FIG. 11

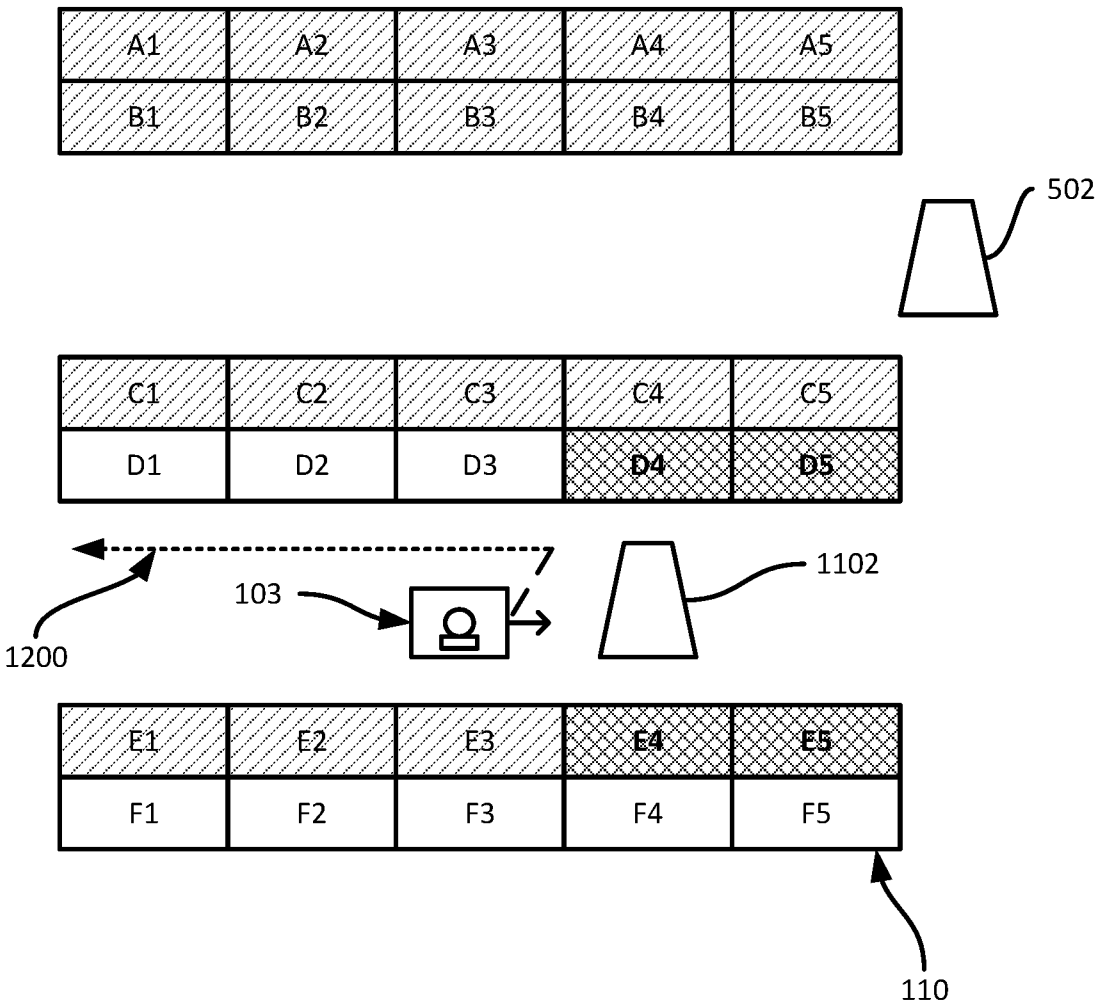

A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
502
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
103
1102
1200
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5
110

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 20/27948

A. CLASSIFICATION OF SUBJECT MATTER

IPC - B25J 9/16 (2020.01)

CPC - B25J 9/1661, B25J 9/1664, B25J 9/16

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2006/0064286 A1 (Fink et al.) 23 March 2006 (23.03.2006) entire document (especiacialy Fig. 1A, Abstract & para [0003], [0008]-[0010], [0014], [0022], [0044], [0045], [0055], [0058], [0065], claim 20).	1-21
Y	US 2014/0195095 A1 (Elbit Systems LTD.) 10 July 2014 (10.07.2014) entire document (especiacialy Fig. 2 & para [0003], [0011], [0029], 0062], [0079], [0080], [0083], [0084], [0089], [0117]).	1-21
Y	US 2015/0355639 A1 (Battelle Energy Alliance, LLC) 10 December 2015 (10.12.2015) entire document (especiacialy para [0216]-[0219], [0292]).	5, 6, 14, 15
Y	US 2018/0321692 A1 (General Electric Company) 08 November 2018 (08.11.2018) entire document (especiacialy para [0027], [0028], [0050]).	9, 18
A	US 2019/0049962 A1 (RavenOPS, Inc.) 14 February 2019 (14.02.2019) entire document.	1-21

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 June 2020 (24.06.2020)	16 JUL 2020
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Lee Young Telephone No. PCT Helpdesk: 571-272-4300