US 20200117214A1

(19) **United States**

(12) **Patent Application Publication**
Jonak et al.

(10) **Pub. No.: US 2020/0117214 A1**

(43) **Pub. Date: Apr. 16, 2020**

(54) **AUTONOMOUS MAP TRAVERSAL WITH WAYPOINT MATCHING**

(71) Applicant: **Boston Dynamics, Inc.**, Waltham, MA (US)

(72) Inventors: **Dom Jonak**, Waltham, MA (US); **Marco da Silva**, Arlington, MA (US); **Joel Chestnutt**, Waltham, MA (US); **Matt Klingensmith**, Waltham, MA (US)

(73) Assignee: **Boston Dynamics, Inc.**, Waltham, MA (US)

(21) Appl. No.: **16/295,592**

(22) Filed: **Mar. 7, 2019**

Related U.S. Application Data

(60) Provisional application No. 62/745,013, filed on Oct. 12, 2018.

Publication Classification

(51) **Int. Cl.**

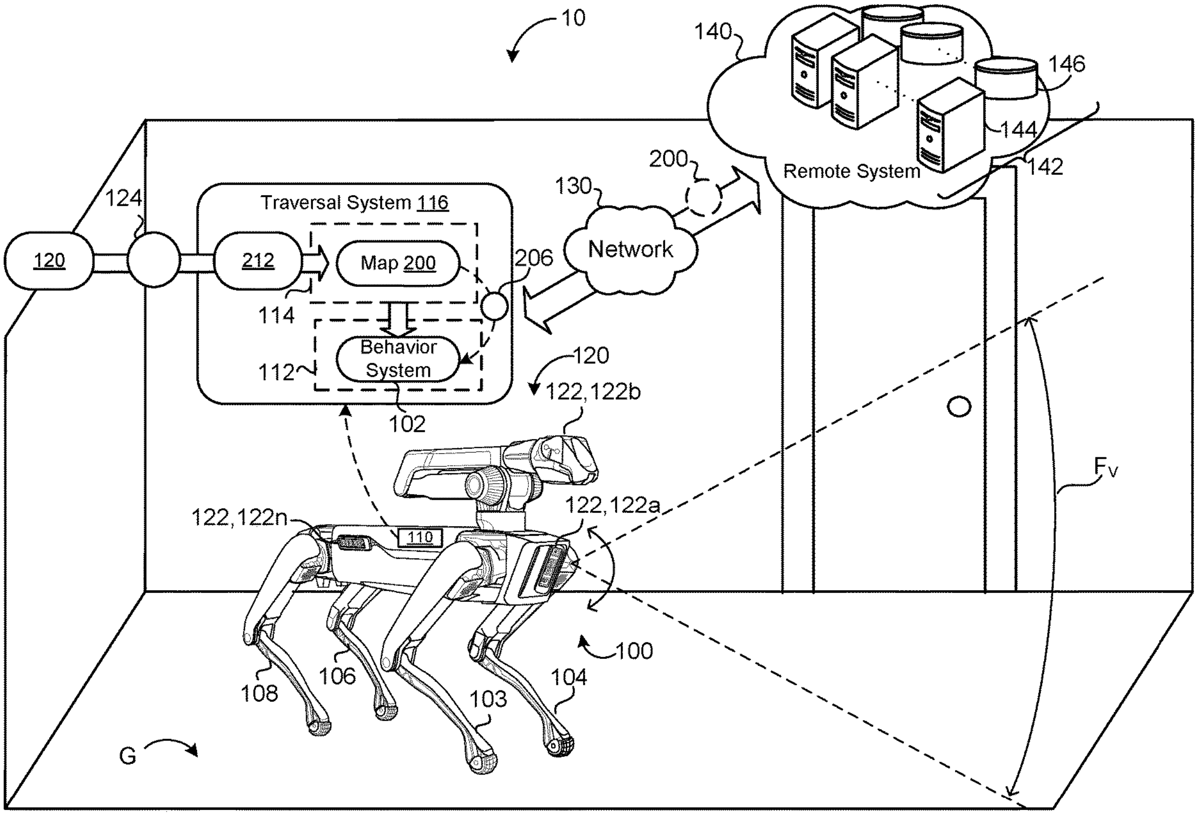
G05D 1/02	(2006.01)
G06T 7/521	(2006.01)
G06T 7/593	(2006.01)
G06T 7/73	(2006.01)
G06T 7/13	(2006.01)
G06K 9/00	(2006.01)
G01S 17/89	(2006.01)

(52) **U.S. Cl.**

CPC ***G05D 1/0274*** (2013.01); ***G06T 7/521*** (2017.01); ***G06T 7/593*** (2017.01); ***G06T 7/74*** (2017.01); ***G06T 7/13*** (2017.01); ***G06K 9/00664*** (2013.01); *G05D 1/0214* (2013.01); ***G01S 17/89*** (2013.01); ***G05D 1/0251*** (2013.01); *G06T 2207/10028* (2013.01); *G06T 2207/10012* (2013.01); *G05B 2219/40298* (2013.01); ***G05D 1/027*** (2013.01)

(57) **ABSTRACT**

A robot includes a drive system configured to maneuver the robot about an environment and data processing hardware in communication with memory hardware and the drive system. The memory hardware stores instructions that when executed on the data processing hardware cause the data processing hardware to perform operations. The operations include receiving image data of the robot maneuvering in the environment and executing at least one waypoint heuristic. The at least one waypoint heuristic is configured to trigger a waypoint placement on a waypoint map. In response to the at least one waypoint heuristic triggering the waypoint placement, the operations include recording a waypoint on the waypoint map where the waypoint is associated with at least one waypoint edge and includes sensor data obtained by the robot. The at least one waypoint edge includes a pose transform expressing how to move between two waypoints.



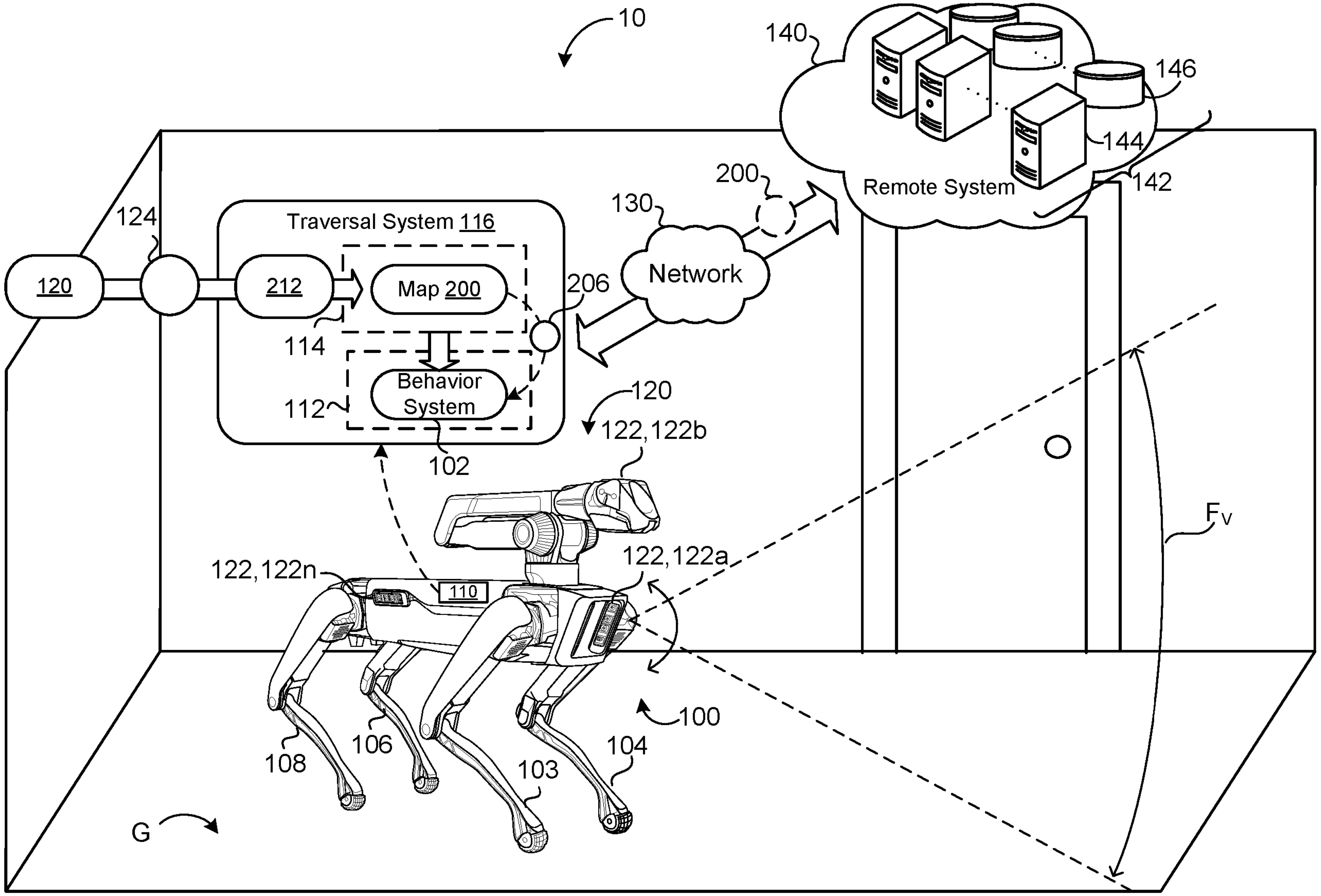

10
140
146
144
142
Remote System
200
130
Network
206
124
Traversal System 116
120
212
Map 200
114
Behavior System
112
102
120
122,122b
122,122n
110
122,122a
Fv
100
108
106
103
104
G

FIG. 1

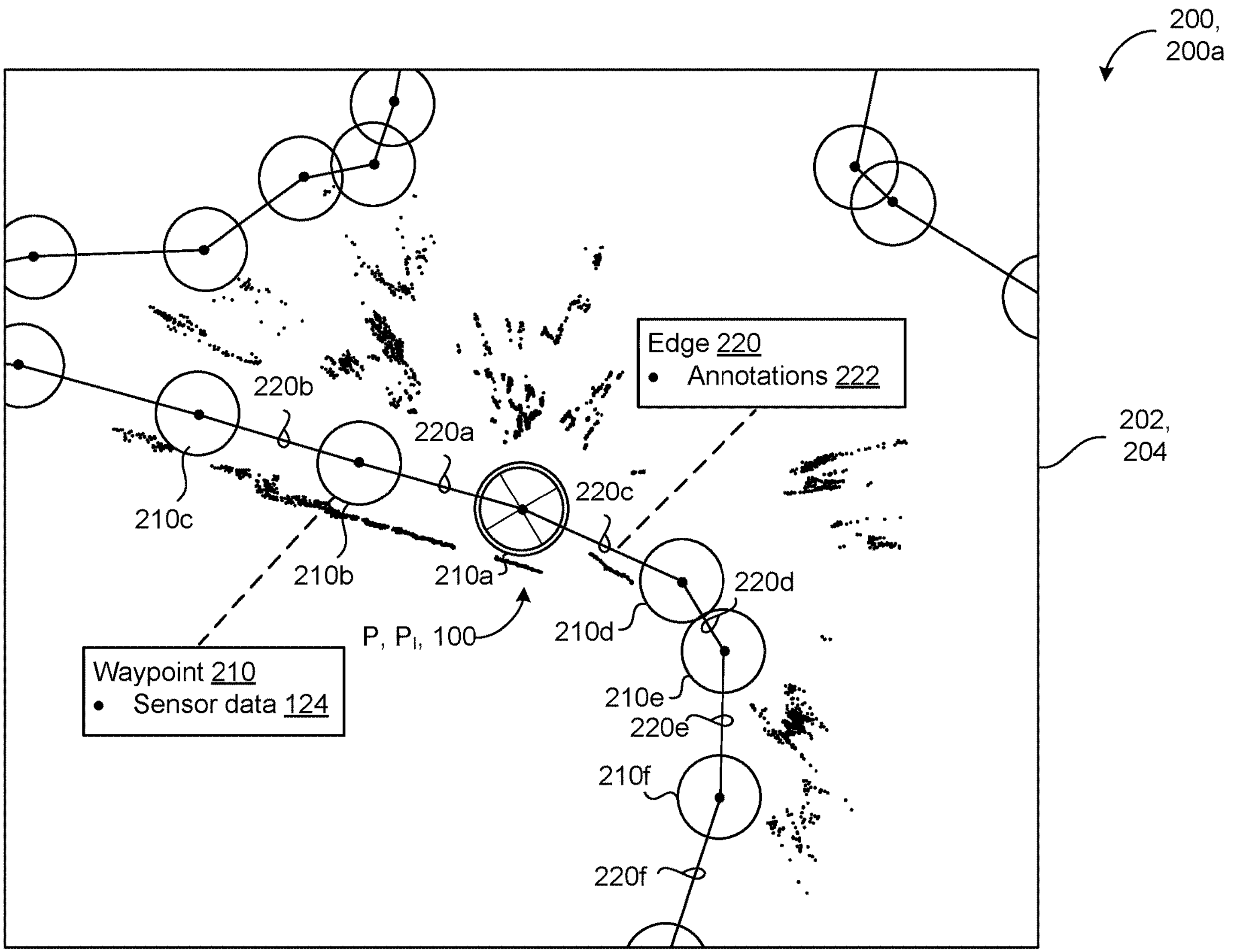

200,
200a
202,
204
Edge 220
• Annotations 222
220b
220a
220c
210c
210b
210a
220d
P, P$_I$, 100
210d
210e
220e
210f
220f
Waypoint 210
• Sensor data 124

FIG. 2A

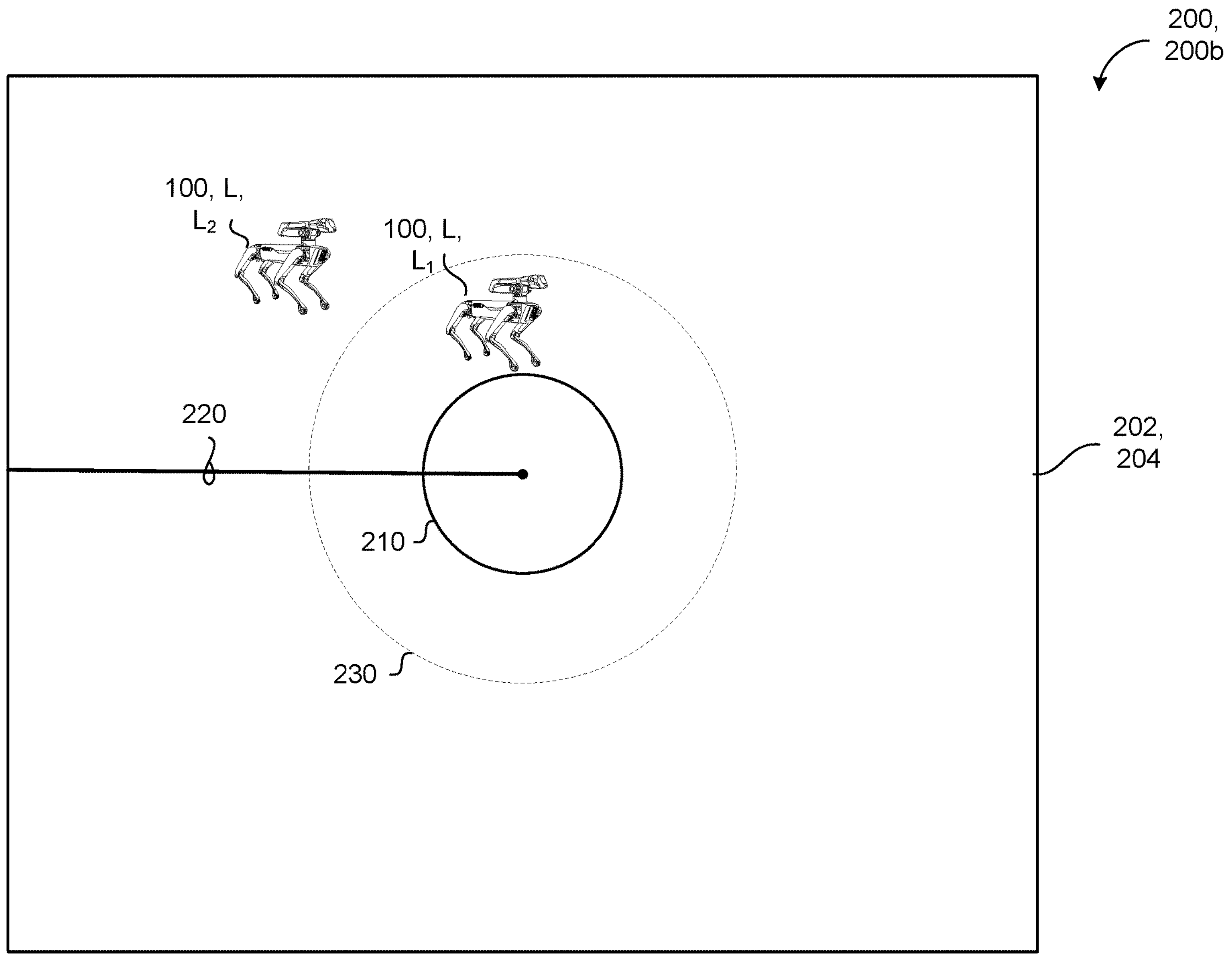


FIG. 2B

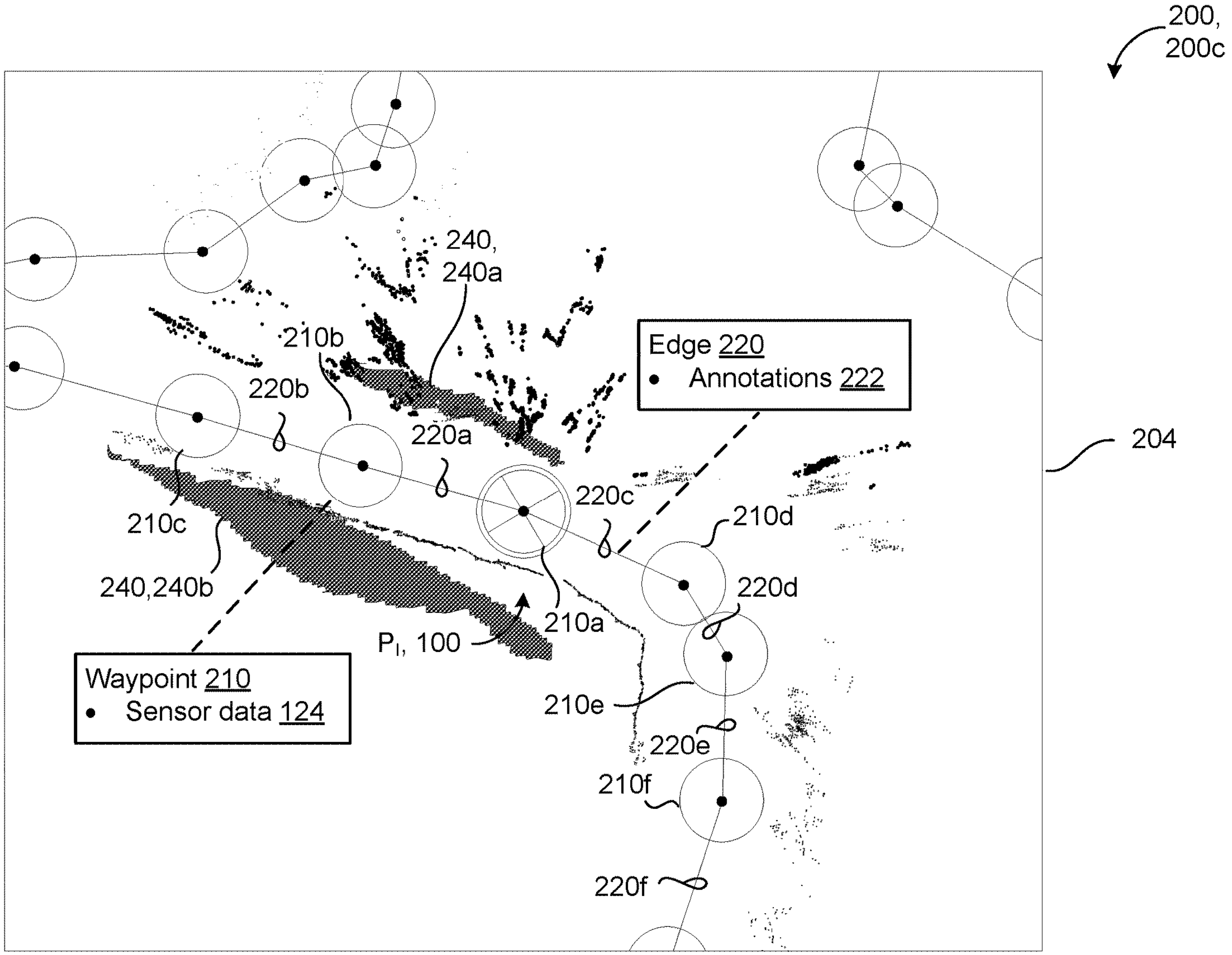

200,
200c
240,
240a
Edge 220
• Annotations 222
210b
220b
220a
204
220c
210d
210c
240,240b
220d
210a
P_l, 100
Waypoint 210
• Sensor data 124
210e
220e
210f
220f

FIG. 2C

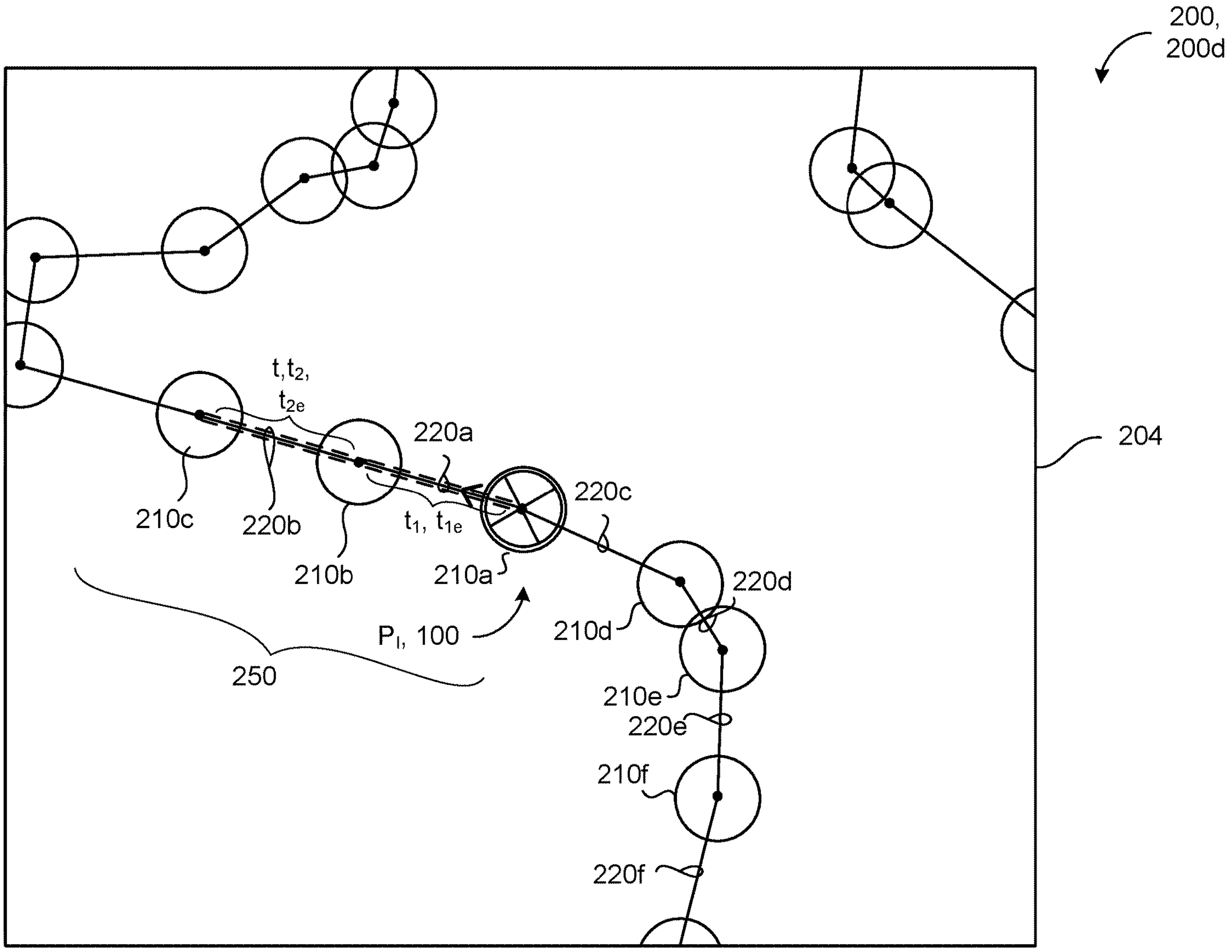

200,
200d
204
t,t2,
t2e
220a
220c
210c
220b
210b
t1, t1e
210a
220d
210d
Pl, 100
250
210e
220e
210f
220f

FIG. 2D

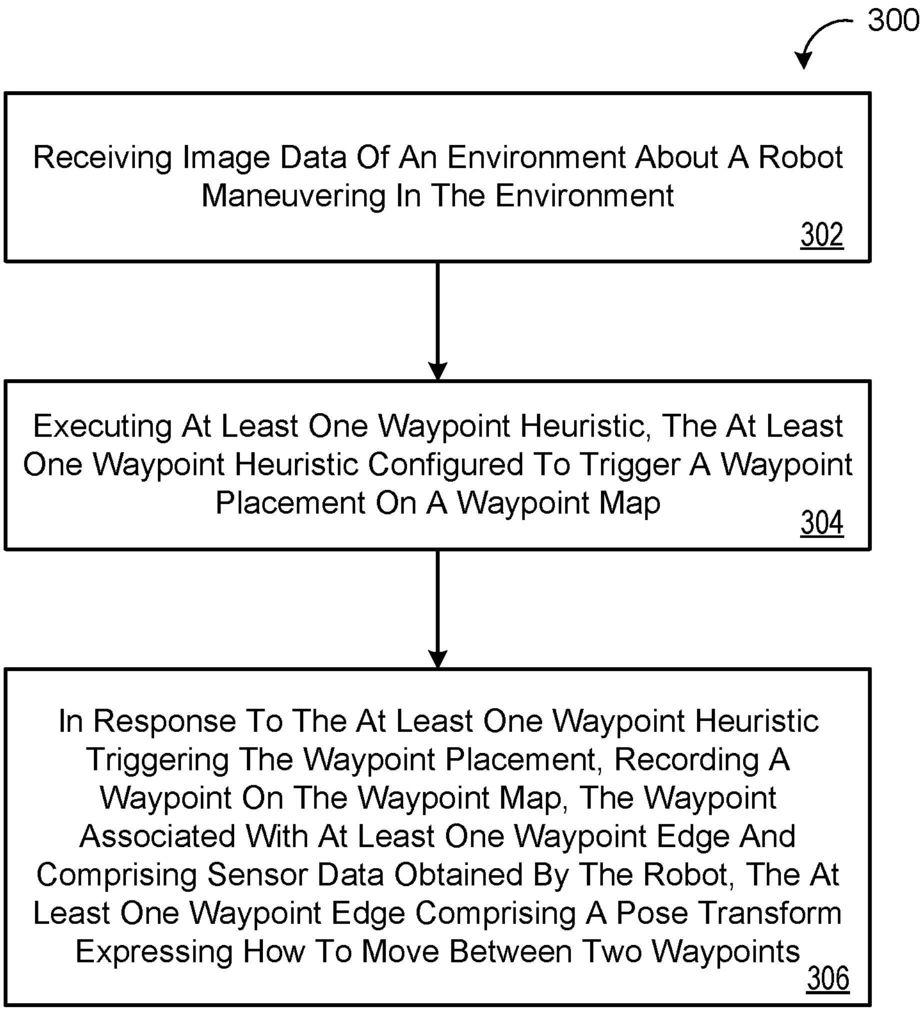

300
Receiving Image Data Of An Environment About A Robot Maneuvering In The Environment
302
Executing At Least One Waypoint Heuristic, The At Least One Waypoint Heuristic Configured To Trigger A Waypoint Placement On A Waypoint Map
304
In Response To The At Least One Waypoint Heuristic Triggering The Waypoint Placement, Recording A Waypoint On The Waypoint Map, The Waypoint Associated With At Least One Waypoint Edge And Comprising Sensor Data Obtained By The Robot, The At Least One Waypoint Edge Comprising A Pose Transform Expressing How To Move Between Two Waypoints
306

FIG. 3

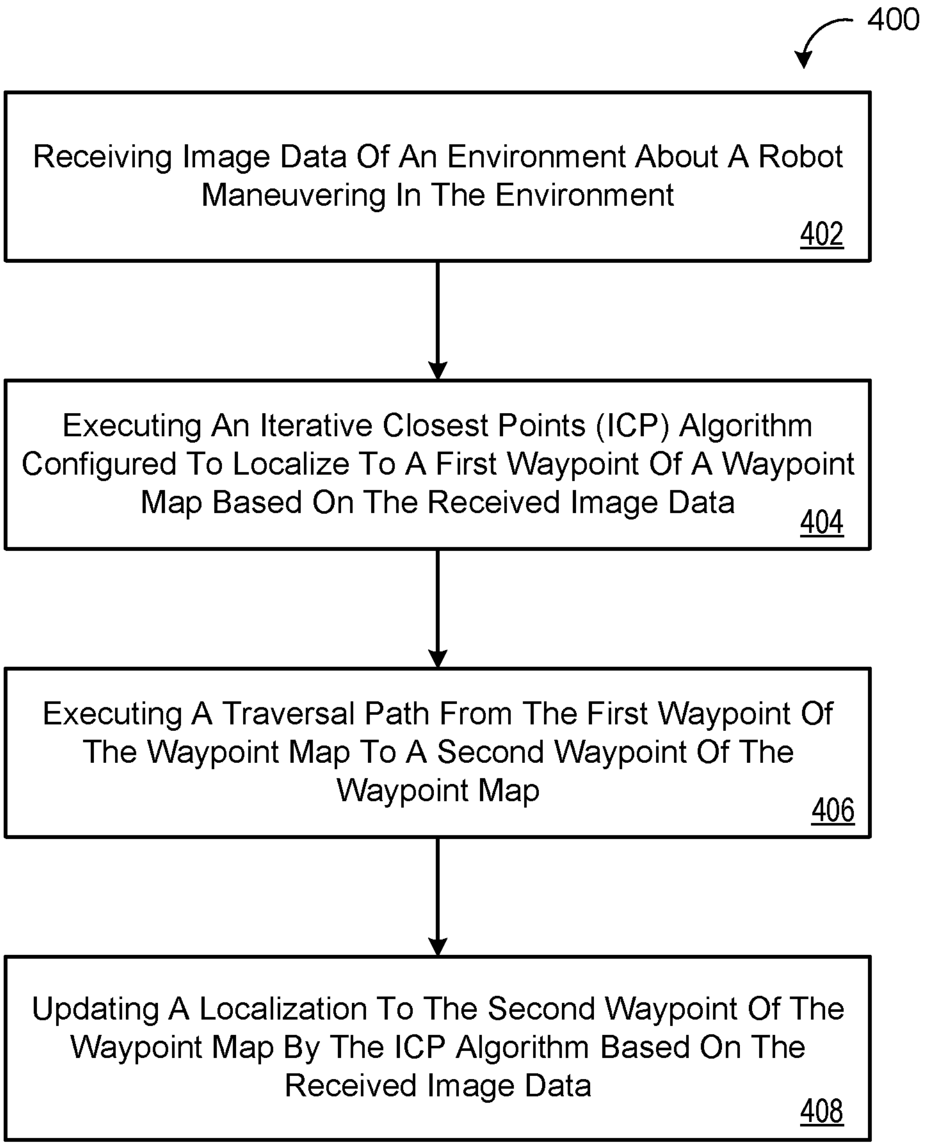

400
Receiving Image Data Of An Environment About A Robot Maneuvering In The Environment
402
Executing An Iterative Closest Points (ICP) Algorithm Configured To Localize To A First Waypoint Of A Waypoint Map Based On The Received Image Data
404
Executing A Traversal Path From The First Waypoint Of The Waypoint Map To A Second Waypoint Of The Waypoint Map
406
Updating A Localization To The Second Waypoint Of The Waypoint Map By The ICP Algorithm Based On The Received Image Data
408

FIG. 4

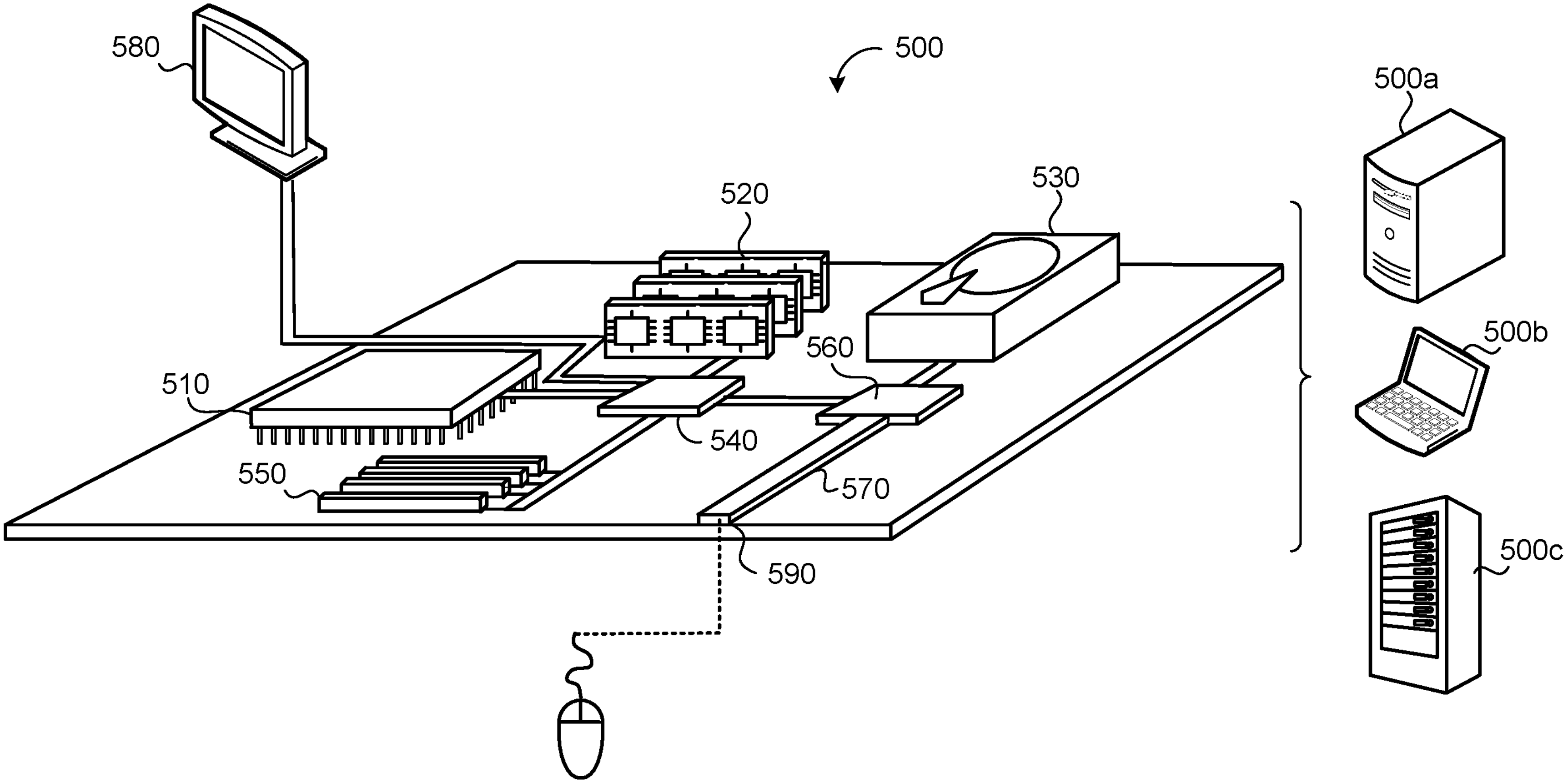


FIG. 5

AUTONOMOUS MAP TRAVERSAL WITH WAYPOINT MATCHING

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This U.S. patent application claims priority under 35 U.S.C. § 119(e) to U.S. Provisional Application 62,745,013, filed on Oct. 12, 2018, which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] This disclosure relates to autonomous map traversal with waypoint matching.

BACKGROUND

[0003] A robot is generally defined as a reprogrammable and multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for a performance of tasks. Robots may be manipulators that are physically anchored (e.g., industrial robotic arms), mobile robots that move throughout an environment (e.g., using legs, wheels, or traction based mechanisms), or some combination of a manipulator and a mobile robot. Robots are utilized in a variety of industries including, for example, manufacturing, transportation, hazardous environments, exploration, and healthcare. As such, the ability to operate autonomously may enhance a robots functionality and provide additional benefits to such industries.

SUMMARY

[0004] One aspect of the disclosure provides a method for autonomous map traversal. The method includes receiving, at data processing hardware, sensor data of an environment about a robot maneuvering in the environment. The sensor data includes image data. The method also includes executing, by the data processing hardware, at least one waypoint heuristic based on the image data. The at least one waypoint heuristic is configured to trigger a waypoint placement on a waypoint map. In response to the at least one waypoint heuristic triggering the waypoint placement, the method includes recording, by the data processing hardware, a waypoint on the waypoint map. The waypoint is associated with at least one waypoint edge and includes at least some of the sensor data obtained by the robot. The at least one waypoint edge includes a pose transform expressing how to move between two waypoints.

[0005] Implementations of the disclosure may include one or more of the following optional features. In some implementations, the image data includes three-dimensional point cloud data captured by a three-dimensional volumetric image sensor. Here, the three-dimensional volumetric image sensor may include one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor. The waypoint heuristic may be configured to detect a threshold feature detection within the image data at a current location of the robot.

[0006] In some examples, the sensor data of the waypoint includes three-dimensional point cloud data captured by a three-dimensional volumetric image sensor and pose data including inertial measurement data measured by an inertial measurement unit (IMU) of the robot. For a given waypoint, the method may include identifying neighboring waypoints to the given waypoint and identifying three-dimensional points of the three-dimensional point cloud data of the given waypoint and the neighboring waypoints as a collection of three-dimensional points. For a given waypoint, the method may also include determining whether the three-dimensional points of the collection of three-dimensional points correspond to a visual edge of an image represented by the three-dimensional point cloud data and associating with the given waypoint the three-dimensional points of the collection of three-dimensional points that correspond to the visual edge of the image represented by the three-dimensional point cloud data.

[0007] In some configurations, the waypoint includes a robot pose constraint configured to cause the robot to achieve an oriented pose at the waypoint. The waypoint map may be locally consistent with neighboring waypoints. The waypoint edge may include an annotation indicating a spatial feature of the environment. Recording the waypoint on the waypoint map include recording a basin zone around the waypoint, the basin zone designating an area adjacent to the waypoint where a robot is configured to use an iterative closest points (ICP) algorithm.

[0008] Another aspect of the disclosure provides a robot. The robot includes a drive system configured to maneuver the robot about an environment and data processing hardware in communication with the drive system. The robot also includes memory hardware in communication with the data processing hardware. The memory hardware stores instructions that when executed on the data processing hardware cause the data processing hardware to perform operations. The operations include receiving sensor data of the robot maneuvering in the environment. The sensor data includes image data. The operations further include executing at least one waypoint heuristic based on the image data. The at least one waypoint heuristic is configured to trigger a waypoint placement on a waypoint map. In response to the at least one waypoint heuristic triggering the waypoint placement, the operations include recording a waypoint on the waypoint map. The waypoint is associated with at least one waypoint edge and includes sensor data obtained by the robot. The at least one waypoint edge includes a pose transform expressing how to move between two waypoints.

[0009] Implementations of the disclosure may include one or more of the following optional features. In some examples, the image data includes three-dimensional point cloud data captured by a three-dimensional volumetric image sensor. Here, the image sensor includes one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor. The waypoint heuristic may be configured to detect a threshold feature detection within the image data at a current location of the robot.

[0010] In some implementations, the sensor data of the waypoint includes three-dimensional point cloud data captured by a three-dimensional volumetric image sensor and pose data including inertial measurement data measured by an inertial measurement unit (IMU) of the robot. For a given waypoint, the operations may include identifying neighboring waypoints to the given waypoint and identifying three-dimensional points of the three-dimensional point cloud data of the given waypoint and the neighboring waypoints as a collection of three-dimensional points. For a given waypoint, the operations may also include determining whether the three-dimensional points of the collection of three-dimensional points correspond to a visual edge of an image represented by the three-dimensional point cloud data and associating with the given waypoint the three-dimensional points of the collection of three-dimensional points that correspond to the visual edge of the image represented by the three-dimensional point cloud data.

[0011] In some configurations, the waypoint includes a robot pose constraint configured to cause the robot to achieve an oriented pose at the waypoint. The waypoint map may be locally consistent with neighboring waypoints. Additionally or alternatively, the waypoint edge may include an annotation indicating a spatial feature of the environment. Recording the waypoint on the waypoint map may include recording a basin zone around the waypoint, the basin zone designating an area adjacent to the waypoint where a robot is configured to use an iterative closest points (ICP) algorithm.

[0012] Yet another aspect of the disclosure provides a method for autonomous map traversal. The method includes receiving, at data processing hardware, image data of an environment about a robot maneuvering in the environment. The method also includes executing, by the data processing hardware, an iterative closest points (ICP) algorithm configured to localize to a first waypoint of a waypoint map based on the received image data. The method further includes executing, by the data processing hardware, a traversal path from the first waypoint of the waypoint map to a second waypoint of the waypoint map and updating, by the data processing hardware, a localization to the second waypoint of the waypoint map by the ICP algorithm based on the received image data.

[0013] Implementations of this aspect of the disclosure may include one or more of the following optional features. In some configurations, the image data includes three-dimensional point cloud data captured by a three-dimensional volumetric image sensor. The image sensor may include one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

[0014] In some examples, updating the localization to the second waypoint includes comparing the received image data with stored image data of the waypoint map, the stored image data associated with the second waypoint. Additionally or alternatively, updating the localization to the second waypoint may include determining that a location associated with the localization satisfies a localization distance threshold and executing, by the data processing hardware, localization at the location according to kinematic data and inertial measurement data. Here, the localization distance threshold indicates a distance from a respective waypoint.

[0015] Another aspect of the disclosure provides a robot. The robot includes a body, legs coupled to the body and configured to maneuver the robot about an environment, and data processing hardware in communication with the legs. The robot also includes memory hardware in communication with the data processing hardware. The memory hardware stores instructions that when executed on the data processing hardware cause the data processing hardware to perform operations. The operations include receiving image data of an environment about a robot maneuvering in the environment. The operations also include executing an iterative closest points (ICP) algorithm configured to localize to a first waypoint of a waypoint map based on the received image data and executing a traversal path from the first waypoint of the waypoint map to a second waypoint of the waypoint map. The operations further include updating a localization to the second waypoint of the waypoint map by the ICP algorithm based on the received image data.

[0016] Implementations of this aspect of the disclosure may include one or more of the following optional features. In some implementations, the image data includes three-dimensional point cloud data captured by a three-dimensional volumetric image sensor. Here, the image sensor includes one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

[0017] In some examples, updating the localization to the second waypoint includes comparing the received image data with stored image data of the waypoint map, the stored image data associated with the second waypoint. Optionally, updating the localization to the second waypoint may also include determining that a location associated with the localization satisfies a localization distance threshold and executing, by the data processing hardware, localization at the location according to kinematic data and inertial measurement data. Here, the localization distance threshold indicates a distance from a respective waypoint.

[0018] The details of one or more implementations of the disclosure are set forth in the accompanying drawings and the description below. Other aspects, features, and advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0019] FIG. **1** is a schematic view of an example of a robot in a robotic environment.

[0020] FIG. **2**A is a schematic view of an example of a map for the robot.

[0021] FIG. **2**B is a schematic view of an example waypoint of a map with a basin.

[0022] FIG. **2**C is a schematic view of example map with map constraints.

[0023] FIG. **2**D is a schematic view of an example map for the robot.

[0024] FIGS. **3** and **4** are flow diagrams of example arrangements for implementing robotic environment traversal for the robot.

[0025] FIG. **5** is a schematic view of an example computing device that may be used to implement the systems and methods described herein.

[0026] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0027] FIG. **1** is an example of a robotic environment **10**. The robotic environment **10** generally refers to a spatial area associated with some type of terrain that may be traversed by a robot **100** according to a map **200**. For example, the terrain is an urban terrain, such as buildings, streets, sidewalks, parks, alleys, stairs, ramps, tunnels, etc., a rural terrain, such as roads, fields, hills, mountains, caves, etc., a subterranean terrain, such as caves, tunnels, etc., or any combination thereof. The robot **100** includes locomotion based structures such as legs and/or wheels attached to a body that enable the robot **100** to move about the robotic environment **10**. Although FIG. **1** depicts a quadruped robot with four legs **103**, **104**, **106**, **108**, the robot **100** may include any number of legs or locomotive based structures (e.g., a biped or humanoid robot with two legs) that provides a means to traverse the robotic environment **10**.

[0028] In some implementations, the robot **100** includes computing hardware **110** and at least one sensor system **120**. The computing hardware **110** generally includes data processing hardware **112** and memory hardware **114**. The computing hardware **110** is configured to control a robot traversal system **116**. The robot traversal system **116** operates a behavior system **102** of the robot **100** to move the robot **100** about the robotic environment **10**. The behavior system **102** is generally responsible for controlling (i.e. executing) behaviors of the robot **100**. For example, the behavior system **102** controls different footstep patterns, leg patterns, body movement patterns, or vision system sensing patterns. The robot traversal system **116** operates the behavior system **102** based on at least one map **200** provided to the robot traversal system **116**.

[0029] The traversal system **116** is configured to communicate with the memory hardware **114** of the robot **100** to provide the map **200** to operate the behavior system **102** using the data processing hardware **112** of the robot **100**. In some examples, the memory hardware **114** stores the map **200** locally on the robot **100**. In other examples, the map **200** is stored and/or accessed remotely by the traversal system **116**. For example, the traversal system **116** communicates via a network **130** with a remote system **140**. The remote system **140** may be a server or cloud-based environment that includes remote resources **142** such as remote data processing hardware **144** and remote memory hardware **146**. In some implementations, the map **200** is stored and/or processed on the remote system **140** using remote resources **142** and is communicated to the traversal system **116** of the robot **100** via the network **130**. In yet other examples, different parts of the map **200** are processed and/or stored remotely (e.g., via the remote system **140**) and locally (e.g., via the computing hardware **110**).

[0030] The sensor system **120** includes one or more sensors **122**, **122***a-n*. The sensors **122** may include vision/image sensors, inertial sensors (e.g., an inertial measurement unit (IMU)), and/or kinematic sensors. Some examples of sensors **122** include a camera such as a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor. In some examples, the sensor **122** has a corresponding field(s) of view F_v defining a sensing range or region corresponding to the sensor **122**. For instance, FIG. **1** depicts a field of a view Fv for the robot **100**. Each sensor **122** may be pivotable and/or rotatable such that the sensor **122** may, for example, change the field of view Fv about one or more axis (e.g., an x-axis, a y-axis, or a z-axis in relation to a ground plane G). In some implementations, the body of the robot **100** includes a sensor system **120** with multiple sensors **122** about the body to gather sensor data **124** in all directions around the robot **100**. For example, the quadruped robot depicted in FIG. **1** includes a sensor **122** on its chest **122***a*, a sensor **122** on each side **122***n* (only one side depicted), and a sensor **122** on its retractable hand **122***b*. The robot **100** may include any number of sensors **122** as part of the sensor system **120** in order to generate sensor data **124** for the robotic environment **10** about the robot **100**.

[0031] When surveying a field of view Fv with a sensor **122**, the sensor system **120** generates sensor data **124** corresponding to the field of view Fv. In some examples, the sensor data **124** is image data that corresponds to a three-dimensional volumetric point cloud generated by a three-dimensional volumetric image sensor **122**. Additionally or alternatively, when the robot **100** is maneuvering about the robotic environment **10**, the sensor system **120** gathers pose data for the robot **100** that includes inertial measurement data (e.g., measured by an IMU). In some examples, the pose data includes kinematic data and/or orientation data about the robot **100**.

[0032] Sensor data **124** gathered by the sensor system **120**, such as the image data, pose data, inertial data, kinematic data, etc., relating to the robotic environment **10** may be communicated to the computing hardware **110** (e.g., the data processing hardware **112** and memory hardware **114**) of the robot **100**. In some examples, the sensor system **120** gathers and stores the sensor data **124** (e.g., in the memory hardware **114** or memory hardware **146** of remote resources **142**). In other examples, the sensor system **120** gathers the sensor data **124** in real-time and processes the sensor data **124** without storing raw (i.e., unprocessed) sensor data **124**. In yet other examples, the computing hardware **110** and/or remote resources **142** store both the processed sensor data **124** and raw sensor data **124**.

[0033] FIGS. **2**A-**2**D are examples of maps **200**, **2001**-*d* generated by the robot **100** and/or remote system **140**. Each map **200** is a representation of an environment (e.g., the robotic environment **10**) used to autonomously guide the robot **100**. The robot **100** (or the remote system **140**) generates the map **200** by receiving the image data **124** of the robotic environment **10** captured by one or more vision sensors **122**, **122***a-c* of the sensor system **120** as the robot **100** maneuvers about the robotic environment **10** as shown in FIG. **1**. In some examples, the robot **100** generates the map **200** with computing hardware **110**, while in other examples, the remote system **140** generates the map **200** with remote resources **142**. Once either the computing hardware **110** or the remote system **140** generates the map **200**, the map **200** is communicated to the traversal system **116** to execute the behavior system **102** of the robot **100**. Here, the behavior system **102** controls the robot **100** to traverse within the robotic environment **10** according to the map **200**. The map **200** may be generated by the same robot **100** that traverses the robotic environment **10** or a different robot.

[0034] Some advantages of the map **200** may be used immediately, repeatedly, and/or transferrably for another robot with a similar sensor system **120**. Here, the map **200** requires no final optimization once all the sensor data **124** has been collected. Furthermore, the robot **100** may autonomously navigate the robotic environment **10** using the map **200** without global positioning information or other navigation data from a beacon. Each map **200** may be automatically processed by the systems of the robot **100** (or the remote system **140**) or manually processed (e.g., editing waypoints **210**, edges **220**, basins **230** (FIG. **2**B), constraints **240** (FIG. **2**C), etc.). This enables the map **200** to be flexible to include additional data (e.g., annotations **222**) or to provide adjusted robotic behaviors (e.g., suppressing ICP localization in feature-less areas). Moreover, modifying waypoints **210** or edges **220** of the map **200** (e.g., by adding, editing, or removing) should have no effect on non-modified portions of the map **200**, and thus, simplifying quality control.

[0035] The map **200** includes waypoints **210** and edges **220** (also referred to as waypoint edges) forming connections between waypoints **210**. A waypoint **210** is a representation of what the robot **100** sensed (e.g., according to its sensor system **120**) at a particular place within the robotic environment **10**. The robot **100** and/or remote system **140** generates waypoints **210** based on the image data **124** collected by the sensor system **120** of the robot. Because the map **200** is generated by waypoints **210**, the map **200** may be locally consistent (e.g., spatially consistent within an area due to neighboring waypoints), but does not need to be globally accurate and/or consistent.

[0036] With the image data **124**, the robot **100** and/or remote system **140** executes at least one waypoint heuristic **212** (FIG. **1**) (e.g., waypoint algorithm). The at least one waypoint heuristic **212** is configured to trigger a waypoint placement on the map **200**. For example, the waypoint heuristic **212** is configured to detect a threshold feature detection within the image data **124** at a location of the robot **100** (e.g., when generating or updating the map **200**). The waypoint heuristic **212** may identify features within the robotic environment **10** that function as reliable vision sensor features offering repeatability for the robot **100** to maneuver about the robotic environment **10**. For instance, the waypoint heuristic **212** is pre-programmed for feature recognition (e.g., programmed with stored features) or programmed to identify features where spatial clusters of volumetric image data **124** occur (e.g., corners of rooms or edges of walls). In response to the at least one waypoint heuristic **212** triggering the waypoint placement, the robot **100** and/or remote system **140** records the waypoint **210** on the map **200**. This waypoint identification process may be repeated as the robot **100** is manually driven through an area (e.g., the robotic environment **10**). In some examples, the robot **100** and/or remote system **140** stores a plurality of maps **200** generated for areas that the robot **100** has previously sensed (via its sensor system **120**).

[0037] Each waypoint **210** is generally associated with a waypoint edge **220**. More specifically, an edge **220** is configured to indicate how one waypoint **210** (e.g., a first waypoint **210***a*) is related to another waypoint **210** (e.g., a second waypoint **210***b*). For example, the edge **220** represents a positional relationship between waypoints **210** (e.g., adjacent waypoints **210**). In other words, the edge **220** is a connection between two waypoints **210** (e.g., the edge **220***a* shown in FIG. **2**A as a connection between a first waypoint **210***a* and a second waypoint **210***b*). For instance, the edge **220** is a path (e.g., a movement path for the robot **100**) between the first waypoint **210***a* to the second waypoint **210***b*. In some examples, each edge **220** includes a pose transform that expresses how the robot **100** moves along the edge **220** between two waypoints **210**. Pose transformations may describe a position and/or orientation of one coordinate frame within an environment relative to another coordinate frame. In some implementations, the edge **220** includes a full three-dimensional transform (e.g., six numbers). Some of these numbers include various estimates such as, for example, a dead reckoning pose estimation, a vision based estimation, or other estimations based on kinematics and/or inertial measurements of the robot **100**.

[0038] In some examples, the edge **220** includes annotations **222** associated with the edge **220** that provide further indication/description of the robotic environment **10**. Some examples of annotations **222** include a description or an indication that an edge **220** is located on stairs or crosses a doorway. These annotations **222** may aid the robot **100** during maneuvering especially when visual information is missing or lacking (e.g., a void such as a doorway). In some configurations, the annotations **222** include directional constraints (also may be referred to as pose constraints). A directional constraint of the annotation **222** may specify an alignment and/or an orientation (e.g., a pose) of the robot **100** at a particular environment feature. For example, the annotation **222** specifies a particular alignment or pose for the robot **100** before traveling along stairs or down a narrow corridor which may restrict the robot **100** from turning.

[0039] In some implementations, each waypoint **210** of the map **200** also includes sensor data **124** corresponding to data collected by the sensor system **120** of the robot **100** when the robot **100** sensed the robotic environment forming the map **200**. Here, the sensor data **124** at a waypoint **210** enables the robot **100** to localize by comparing real-time sensor data **124** gathered as the robot **100** traverses the robotic environment **10** according to the map **200** with sensor data **124** stored for the waypoints **210** of the map **200**. In some configurations, after the robot **100** moves along an edge **220** (e.g., with the intention to be at a target waypoint **210**), the robot **100** is configured to localize by directly comparing real-time sensor data **124** with the map **200** (e.g., sensor data **124** associated with the intended target waypoint **210** of the map **200**). Here, by storing raw or near-raw sensor data **124** with minimal processing for the waypoints **210** of the map **200**, the robot **100** may use real-time sensor data **124** to localize efficiently as the robot **100** maneuvers within the mapped robotic environment **10**. In some examples, an iterative closest points (ICP) algorithm localizes the robot **100** with respect to a waypoint **210**.

[0040] The sensor data **124** may also allow the initialization of the robot **100** to autonomously traverse the robotic environment **10** using the map **200**. In some examples, the robot **100** initially receives a hint defining an initial pose P, P_I (e.g., as shown in FIG. **2**A) of the robot **100** relative to a waypoint **210**. The robot **100** may be configured to receive this hint automatically or from an operator of the robot **100**. Here, the pose P refers to a kinematic position and/or orientation of the robot **100**. With the initial pose P_I of the robot **100** relative to a respective waypoint **210**, the robot **100** traverses the map **200** according to commands from the traverse system **116** that instruct the robot **100** to move along edges **220**. For example, the map **200***a* of FIG. **2**A shows the robot **100** moving between waypoints **210***a-f* along edged **220***a-f.*

[0041] In some configurations, the map **200** includes an unprocessed map **202** and/or a processed map **204**. The unprocessed map **202** is a map **200** that includes all the raw or nearly raw sensor data **124** gathered by the sensor system **120** of the robot **100** during generation of the map **200** (e.g., shown as points and/or point clouds in FIG. **2**A). In contrast, the processed map **204** processes sensor data **124** for a waypoint **210** with immediate neighbors of the waypoint **210**. In other words, a processed map **204** may contain a greater extent of data, such as data for multiple waypoints **210** within the robotic environment **110**. In some examples, the processed map **204** reduces noise near waypoints **210** and edges **220** to increase the effectiveness for autonomous traversal. For example, referring to the map **200***a* of FIG. **2**A, the sensor data **124** for waypoint **210***a* will be processed with all sensor data **124** of neighboring waypoints **210***b,*

210*d*. In some examples, the processed map **204** depicts, as a visual image for the processed map **204**, points (e.g., volumetric points) that correspond to an edge **220**. As an example, for a given waypoint **210***a*, the robot **100** or remote system **140** generating the processed map **204** (i.e., the processing system), identifies neighboring waypoints **210***b*, **210***d* for the given waypoint **210***a*. With the identified neighboring waypoints **210***b*, **210***d*, the traversal system **116** determines whether three-dimensional points of a three-dimensional point cloud corresponding the waypoints **210***a*, **210***b*, **210***d* corresponds to a visual edge of an image represented by the three-dimensional point cloud data. Based on this determination, the traversal system **116** associates with the given waypoint **210***a* the three-dimensional points that correspond to the visual edge of the image represented by the three-dimensional point cloud data. Additionally or alternatively, the processing for the processed map **204** removes any sensor data **124** along an edge **220** that has not been observed by the sensor system **120** from multiple vantage points.

[0042] FIG. 2B is an example of a map **200***b* that uses basin zones around one or more waypoints **210**. Here, a waypoint **210** is configured with a corresponding basin **230**. The basin **230** refers to a designated zone around the waypoint **210** (e.g., shown in FIG. 2B as the area within the dotted circle labeled **230**) where the robot **100** uses the ICP algorithm to determine its movement relative to the waypoint **210**. By using the ICP algorithm, the robot **100** may fine-tune its position. When the robot **100** determines that a location L of the robot **100** is outside of the basin **230** of a corresponding waypoint **210**, the robot **100** is configured to use kinematics and/or inertial measurements to determine the movement of the robot **100**. In the example shown, when the robot **100** is at a first location L_1, the robot **100** uses an ICP algorithm to determine the movement of the robot **100**. Yet when the robot **100** is at a second location L_2, the robot **100** determines the movement of the robot **100** according to kinematics and/or inertial measurements.

[0043] In some examples, the area within the waypoint **210** is referred to as a goal zone. Once the robot **100** is within the goal zone, the robot **100** has successfully navigated to the waypoint **210**. Upon successful navigation to the waypoint **210** (i.e., entry of the goal zone) the robot **100** may proceed to move toward a subsequent waypoint **210** along a path of the robot **100**.

[0044] In some configurations, when a quality of the localization is poor, the traversal system **116** may cease to autonomously guide the robot **100** according to the map **200**. This cessation may occur based on the basin **230** or based on, for example, a localization threshold (e.g., a location L from the waypoint **210** where an operator of the robot **100** would consider the robot **100** lost). For instance, the localization threshold is a distance from a waypoint **210** determined to be outside a range of the waypoint **210** to perform localization based on the ICP algorithm.

[0045] Additionally or alternatively, the robot **100** may utilize global positioning (e.g., a global positioning system (GPS) receiver) while navigating according to kinematics, inertial measurements, or the ICP algorithm. In some examples, the robot **100** first uses a map **200** to determine an initial heading for the robot **100** (e.g., along an edge **220** between waypoints **210**), but subsequently supplements navigation (e.g., supplements kinematics, inertial measurements, or the ICP algorithm) with GPS measurements received at the robot **100** from a GPS receiver. In some configurations, the computing hardware **110** is configured to receive GPS measurements (e.g., the computing hardware **110** is configured with a GPS receiver). A supplemental GPS system may be particularly useful when the robot **100** is navigating in an outdoor robotic environment **10**. The robot **100** may be configured such that the robot **100** always uses the GPS system or utilizes the GPS system to provide navigation inputs when the robot **100** senses a particular robotic environment **10** (e.g., when the robot **100** senses an outdoor environment with less adjacent features). For instance, when the quality of the localization is poor, the robot **100** activates and/or uses the GPS system to provide navigational inputs. In some configurations, the GPS system traces the path of the robot **100**. In these configurations, the GPS path may be compared to the edges **220** on a map **200** to provide feedback to the traversal system **116** (e.g., to fine-tune autonomous navigation using the traversal system **116**).

[0046] Referring to FIG. 2C, in some implementations, a map **200***c* includes map constraints **240**, **240***a-b*. A map constraint **240** refers to a virtual obstacle that prevents the robot **100** from deviating into areas designated or blocked with the virtual obstacle. In other words, a map constraint **240** may help prevent the robot **100** from significantly departing from a movement path corresponding to one or more edges **220**. The map constraint **240** may be placed relative to a recorded path, manually added by an operator, or precomputed based on scene geometry. For example, in FIG. 2C, the edges **220***a-c* may correspond to a corridor. When the robot **100** recorded the map **200** with waypoints **210***a-c* and edges **220***a-c* corresponding to the corridor, the robot **100** may have never been driven into rooms and/or doorways on either side of the corridor. It may be advantageous to block off these rooms at their respective doorways by a first map constraint **240***a* and a second map constraint **240***b*. Accordingly with the first map constraint **240***a* and the second map constraint **240***b*, the robot **100** should be prevented from wandering into the rooms when following the edges **220***a-c*, thereby preventing the robot **100** from becoming lost or off-course.

[0047] FIG. 2D shows a map **200***d* (e.g., a processed map **204**) with a time-indexed robot trajectory **250** that the traversal system **116** configures the robot **100** to follow. The time-indexed robot trajectory **250** is a movement trajectory for the robot **100** between waypoints **210**. The time-indexed robot trajectory **250** may also be referred to as a traversal path **250**. Here, FIG. 2D depicts the time-indexed robot trajectory **250** by an arrowhead pointing from a first waypoint **210***a* where the robot **100** is currently positioned (shown as an “X” within the waypoint **210***a*) towards a third waypoint **210***c*. To navigate the time-indexed robot trajectory **250**, the robot **100** will traverse edges **220***a*, **220***b*.

[0048] In some examples, the traversal system **116** generates a time-indexed robot trajectory **250** across upcoming waypoints **210** for movement of the robot **100**. In some implementations, for the time-indexed robot trajectory **250**, the traversal system **116** specifies that an orientation for the robot **100** at each waypoint **210** is halfway between current and next edges **220** of a route through a respective waypoint **210**. Here, the time t may be based on an expected time to reach each waypoint **210** due to the trajectory. As the robot **100** travels the trajectory **250**, times t (e.g., the time t**2** to travel from a second waypoint **210***b* to the third waypoint

210*c*) for the trajectory **250** may be updated based on updated estimates of when the robot **100** should reach each waypoint **210**. In this configuration, when a time t corresponding to a waypoint **210** is close to a current time, the waypoint's time is updated. In some examples, after the current time passes a time associated with the waypoint **210**, the traversal system **116** considers the waypoint **210** reached and subsequently removes it from the trajectory.

[0049] Additionally or alternatively, as shown in the maps **200***a,* **200***c* of FIGS. **2**A and **2**C, annotations **222** associated with edges **220** may modify an orientation of the robot **100** at waypoints **210** associated with a time-indexed robot trajectory such that the robot **100** conforms with a directional constraint of the annotation **222**. Based on these directional constraints, an orientation of the robot **100** may also be estimated for a waypoint **210** within the trajectory. When an orientation estimate determines that the orientation will be incorrect in light of the directional constraint, an orientation segment (e.g., a turn-in-place segment) may be added to the time-indexed robot trajectory to ensure that the robot **100** complies with the directional constraint.

[0050] Referring back to FIG. **1**, the behavior system **102** of the robot **100** may optionally receive a path description **206** from the traversal system **116** that includes goal conditions and constraint conditions for reaching one or more waypoints **210**. Based on these goal conditions and constraint conditions, the robot **100** follows the path description **206** with an ability to adjust its route as long as the adjusted route satisfies the goal and constraint conditions. Some examples of conditions include specifying a position or orientation for the robot **100** at or around a target waypoint **210** or specifying a position, orientation, and/or speed for travel to the target waypoint **210**.

[0051] FIG. **3** is an example flow diagram of a method **300** to configure a map **200**. At operation **302**, the method **300** receives image data **124** of a robotic environment **10** about a robot **100** maneuvering in the robotic environment **10**. At operation **304**, the method **300** executes at least one waypoint heuristic **212** where the at least one waypoint heuristic **212** is configured to trigger a waypoint placement on a waypoint map **200**. In response to the at least one waypoint heuristic **212** triggering the waypoint placement, at operation **306**, the method **300** records a waypoint **210** on the waypoint map **200**. Here, the waypoint **210** is associated with at least one waypoint edge **220** and includes sensor data **124** obtained by the sensor system **120** of the robot **100**. The at least one waypoint edge **220** may include a pose transform expressing how to move between two waypoints **210**.

[0052] FIG. **4** is an example flow diagram of method **400** to localize the robot **100** with respect to a waypoint map **200**. At operation **402**, the method **400** receives image data **124** of a robotic environment **10** about a robot **100** maneuvering in the robotic environment **10**. At operation **404**, the method **400** executes an iterative closest points (ICP) algorithm configured to localize to a first waypoint **210***a* of the waypoint map **200** based on the received image data **124**. At operation **406**, the method **400** executes a traversal path **250** from the first waypoint **210***a* of the waypoint map **200** to a second waypoint **210***b* of the waypoint map **200**. At operation, **404**, the method **400** updates the localization to the second waypoint **210***b* of the waypoint map **200** by the ICP algorithm based on the received image data.

[0053] FIG. **5** is schematic view of an example computing device **500** that may be used to implement the systems and methods described in this document, for example, the behavior system **102**, the computing hardware **110**, the traversal system **116**, or the remote resources **142**. The computing device **500** is intended to represent various forms of digital computers, such as laptops, desktops, workstations, personal digital assistants, servers, blade servers, mainframes, and other appropriate computers. The components shown here, their connections and relationships, and their functions, are meant to be exemplary only, and are not meant to limit implementations of the inventions described and/or claimed in this document.

[0054] The computing device **500** includes a processor **510**, memory **520**, a storage device **530**, a high-speed interface/controller **540** connecting to the memory **520** and high-speed expansion ports **550**, and a low speed interface/controller **560** connecting to a low speed bus **570** and a storage device **530**. Each of the components **510**, **520**, **530**, **540**, **550**, and **560**, are interconnected using various busses, and may be mounted on a common motherboard or in other manners as appropriate. The processor **510** can process instructions for execution within the computing device **500**, including instructions stored in the memory **520** or on the storage device **530** to display graphical information for a graphical user interface (GUI) on an external input/output device, such as display **580** coupled to high speed interface **540**. In other implementations, multiple processors and/or multiple buses may be used, as appropriate, along with multiple memories and types of memory. Also, multiple computing devices **500** may be connected, with each device providing portions of the necessary operations (e.g., as a server bank, a group of blade servers, or a multi-processor system).

[0055] The memory **520** stores information non-transitorily within the computing device **500**. The memory **520** may be a computer-readable medium, a volatile memory unit(s), or non-volatile memory unit(s). The non-transitory memory **520** may be physical devices used to store programs (e.g., sequences of instructions) or data (e.g., program state information) on a temporary or permanent basis for use by the computing device **500**. Examples of non-volatile memory include, but are not limited to, flash memory and read-only memory (ROM)/programmable read-only memory (PROM)/erasable programmable read-only memory (EPROM)/electronically erasable programmable read-only memory (EEPROM) (e.g., typically used for firmware, such as boot programs). Examples of volatile memory include, but are not limited to, random access memory (RAM), dynamic random access memory (DRAM), static random access memory (SRAM), phase change memory (PCM) as well as disks or tapes.

[0056] The storage device **530** is capable of providing mass storage for the computing device **500**. In some implementations, the storage device **530** is a computer-readable medium. In various different implementations, the storage device **530** may be a floppy disk device, a hard disk device, an optical disk device, or a tape device, a flash memory or other similar solid state memory device, or an array of devices, including devices in a storage area network or other configurations. In additional implementations, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory **520**, the storage device **530**, or memory on processor **510**.

[0057] The high speed controller **540** manages bandwidth-intensive operations for the computing device **500**, while the low speed controller **560** manages lower bandwidth-intensive operations. Such allocation of duties is exemplary only. In some implementations, the high-speed controller **540** is coupled to the memory **520**, the display **580** (e.g., through a graphics processor or accelerator), and to the high-speed expansion ports **550**, which may accept various expansion cards (not shown). In some implementations, the low-speed controller **560** is coupled to the storage device **530** and a low-speed expansion port **590**. The low-speed expansion port **590**, which may include various communication ports (e.g., USB, Bluetooth, Ethernet, wireless Ethernet), may be coupled to one or more input/output devices, such as a keyboard, a pointing device, a scanner, or a networking device such as a switch or router, e.g., through a network adapter.

[0058] The computing device **500** may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a standard server **500***a* or multiple times in a group of such servers **500***a,* as a laptop computer **500***b,* or as part of a rack server system **500***c.*

[0059] Various implementations of the systems and techniques described herein can be realized in digital electronic and/or optical circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

[0060] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms "machine-readable medium" and "computer-readable medium" refer to any computer program product, non-transitory computer readable medium, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term "machine-readable signal" refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0061] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit). Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[0062] To provide for interaction with a user, one or more aspects of the disclosure can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube), LCD (liquid crystal display) monitor, or touch screen for displaying information to the user and optionally a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's client device in response to requests received from the web browser.

[0063] A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the disclosure. Accordingly, other implementations are within the scope of the following claims.

What is claimed is:

1. A method comprising:

receiving, at data processing hardware, sensor data of an environment about a robot maneuvering in the environment, the sensor data comprising image data;

executing, by the data processing hardware, at least one waypoint heuristic based on the image data, the at least one waypoint heuristic configured to trigger a waypoint placement on a waypoint map; and

in response to the at least one waypoint heuristic triggering the waypoint placement, recording, by the data processing hardware, a waypoint on the waypoint map, the waypoint associated with at least one waypoint edge and comprising at least some of the sensor data obtained by the robot, the at least one waypoint edge comprising a pose transform expressing how to move between two waypoints.

2. The method of claim **1**, wherein the image data comprises three-dimensional point cloud data captured by a three-dimensional volumetric image sensor.

3. The method of claim **2**, wherein the three-dimensional volumetric image sensor comprises one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

4. The method of claim **1**, wherein the waypoint heuristic is configured to detect a threshold feature detection within the image data at a current location of the robot.

5. The method of claim **1**, wherein the sensor data of the waypoint comprises:

three-dimensional point cloud data captured by a three-dimensional volumetric image sensor; and

pose data comprising inertial measurement data measured by an inertial measurement unit (IMU) of the robot.

6. The method of claim **5**, further comprising, for a given waypoint:

identifying neighboring waypoints to the given waypoint;

identifying three-dimensional points of the three-dimensional point cloud data of the given waypoint and the neighboring waypoints as a collection of three-dimensional points;

determining whether the three-dimensional points of the collection of three-dimensional points correspond to a visual edge of an image represented by the three-dimensional point cloud data; and

associating with the given waypoint the three-dimensional points of the collection of three-dimensional points that correspond to the visual edge of the image represented by the three-dimensional point cloud data.

7. The method of claim **1**, wherein the waypoint comprises a robot pose constraint configured to cause the robot to achieve an oriented pose at the waypoint.

8. The method of claim **1**, wherein the waypoint map is locally consistent with neighboring waypoints.

9. The method of claim **1**, wherein the waypoint edge comprises an annotation indicating a spatial feature of the environment.

10. The method of claim **1**, wherein recording the waypoint on the waypoint map further comprises recording a basin zone around the waypoint, the basin zone designating an area adjacent to the waypoint where a robot is configured to use an iterative closest points (ICP) algorithm.

11. A robot comprising:

a drive system configured to maneuver the robot about an environment;

data processing hardware in communication with the drive system; and

memory hardware in communication with the data processing hardware, the memory hardware storing instructions that when executed on the data processing hardware cause the data processing hardware to perform operations comprising:

receiving sensor data of the robot maneuvering in the environment, the sensor data comprising image data;

executing at least one waypoint heuristic based on the image data, the at least one waypoint heuristic configured to trigger a waypoint placement on a waypoint map; and

in response to the at least one waypoint heuristic triggering the waypoint placement, recording a waypoint on the waypoint map, the waypoint associated with at least one waypoint edge and comprising sensor data obtained by the robot, the at least one waypoint edge comprising a pose transform expressing how to move between two waypoints.

12. The robot of claim **11**, wherein the image data comprises three-dimensional point cloud data captured by a three-dimensional volumetric image sensor.

13. The robot of claim **12**, wherein the image sensor comprises one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

14. The robot of claim **11**, wherein the waypoint heuristic is configured to detect a threshold feature detection within the image data at a current location of the robot.

15. The robot of claim **11**, wherein the sensor data of the waypoint comprises:

three-dimensional point cloud data captured by a three-dimensional volumetric image sensor; and

pose data comprising inertial measurement data measured by an inertial measurement unit (IMU) of the robot.

16. The robot of claim **15**, wherein the operations further comprise, for a given waypoint:

identifying neighboring waypoints to the given waypoint;

identifying three-dimensional points of the three-dimensional point cloud data of the given waypoint and the neighboring waypoints as a collection of three-dimensional points;

determining whether the three-dimensional points of the collection of three-dimensional points correspond to a visual edge of an image represented by the three-dimensional point cloud data; and

associating with the given waypoint the three-dimensional points of the collection of three-dimensional points that correspond to the visual edge of the image represented by the three-dimensional point cloud data.

17. The robot of claim **11**, wherein the waypoint comprises a robot pose constraint configured to cause the robot to achieve an oriented pose at the waypoint.

18. The robot of claim **11**, wherein the waypoint map is locally consistent with neighboring waypoints.

19. The robot of claim **11**, wherein the waypoint edge comprises an annotation indicating a spatial feature of the environment.

20. The robot of claim **11**, wherein recording the waypoint on the waypoint map further comprises recording a basin zone around the waypoint, the basin zone designating an area adjacent to the waypoint where a robot is configured to use an iterative closest points (ICP) algorithm.

21. A method comprising:

receiving, at data processing hardware, image data of an environment about a robot maneuvering in the environment;

executing, by the data processing hardware, an iterative closest points (ICP) algorithm configured to localize to a first waypoint of a waypoint map based on the received image data;

executing, by the data processing hardware, a traversal path from the first waypoint of the waypoint map to a second waypoint of the waypoint map; and

updating, by the data processing hardware, a localization to the second waypoint of the waypoint map by the ICP algorithm based on the received image data.

22. The method of claim **21**, wherein the image data comprises three-dimensional point cloud data captured by a three-dimensional volumetric image sensor.

23. The method of claim **22**, wherein the image sensor comprises one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

24. The method of claim **21**, wherein updating the localization to the second waypoint further comprises comparing the received image data with stored image data of the waypoint map, the stored image data associated with the second waypoint.

25. The method of claim **21**, wherein updating the localization to the second waypoint further comprises:

determining that a location associated with the localization satisfies a localization distance threshold, the localization distance threshold indicating a distance from a respective waypoint; and

executing, by the data processing hardware, localization at the location according to kinematic data and inertial measurement data.

26. A robot comprising:

a body;

legs coupled to the body and configured to maneuver the robot about an environment;

data processing hardware in communication with the legs; and

memory hardware in communication with the data processing hardware, the memory hardware storing instructions that when executed on the data processing hardware cause the data processing hardware to perform operations comprising:

receiving image data of an environment about a robot maneuvering in the environment;

executing an iterative closest points (ICP) algorithm configured to localize to a first waypoint of a waypoint map based on the received image data;

executing a traversal path from the first waypoint of the waypoint map to a second waypoint of the waypoint map; and

updating a localization to the second waypoint of the waypoint map by the ICP algorithm based on the received image data.

27. The robot of claim **26**, wherein the image data comprises three-dimensional point cloud data captured by a three-dimensional volumetric image sensor.

28. The robot of claim **27**, wherein the image sensor comprises one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

29. The robot of claim **26**, wherein updating the localization to the second waypoint further comprises comparing the received image data with stored image data of the waypoint map, the stored image data associated with the second waypoint.

30. The robot of claim **26**, wherein updating the localization to the second waypoint further comprises:

determining that a location associated with the localization satisfies a localization distance threshold, the localization distance threshold indicating a distance from a respective waypoint; and

executing, by the data processing hardware, localization at the location according to kinematic data and inertial measurement data.

* * * * *