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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
28 October 2021 (28.10.2021)

WIPO | PCT

(10) **International Publication Number**
WO 2021/216235 A1

(51) **International Patent Classification:**
B62D 57/024 (2006.01) *B62D 57/032* (2006.01)

(21) **International Application Number:**
PCT/US2021/022953

(22) **International Filing Date:**
18 March 2021 (18.03.2021)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**

63/012,614	20 April 2020 (20.04.2020)	US
16/877,680	19 May 2020 (19.05.2020)	US

(71) **Applicant: BOSTON DYNAMICS, INC.** [US/US]; 200 Smith Street, Suite 4100, Waltham, MA 02451 (US).

(72) **Inventor: KOMOROSKI, Adam**; 200 Smith Street, Suite 4100, Waltham, MA 02451 (US).

(74) **Agent: KRUEGER, Brett, A.**; Honigman LLP, 650 Trade Centre Way, Suite 200, Kalamazoo, MI 49002-0402 (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, IT, 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.

(54) **Title:** IDENTIFYING STAIRS FROM FOOTFALLS

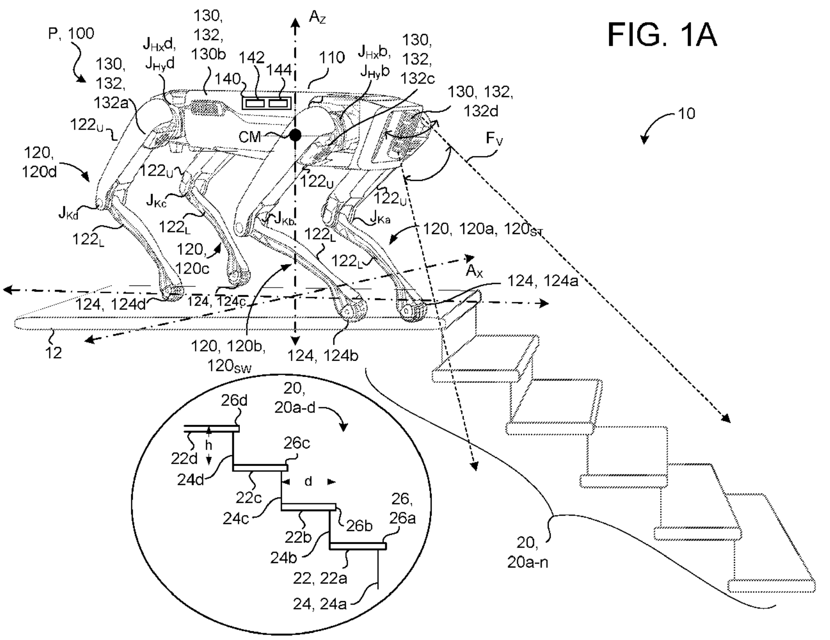


FIG. 1A

(57) **Abstract:** A method of identifying stairs (20) includes receiving a plurality of footfall locations (128) of a robot (100). Each respective footfall location indicates a location where a leg (120) of the robot contacted a support surface (12). The method also includes determining a plurality of candidate footfall location pairs (212) where the candidate footfall location pair includes a first and a second candidate footfall location. The method further includes clustering the first candidate footfall location into a first cluster group (222) based on a height of the first candidate footfall location and clustering the second candidate footfall location into a second cluster group based on a height of the second candidate footfall location. The method additionally includes generating a stair model (202) by representing each of the cluster groups as a corresponding stair and delineating each stair based on a respective midpoint (MP) between each adjacent cluster group.

WO 2021/216235 A1

[Continued on next page]

(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).

Published:

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

Identifying Stairs from Footfalls

TECHNICAL FIELD

[0001] This disclosure relates to identifying stairs from footfalls.

BACKGROUND

[0002] 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 of robots to traverse environments with obstacles or features requiring various means of coordinated leg movement provides additional benefits to such industries.

SUMMARY

[0003] One aspect of the disclosure provides a method of identifying stairs from footfalls. The method includes receiving, at data processing hardware, a plurality of footfall locations of a robot traversing an environment. Here, each respective footfall location indicates a location where a leg of the robot contacted a support surface beneath the robot. The method also includes determining, by the data processing hardware, a plurality of candidate footfall location pairs based on the plurality of footfall locations. The candidate footfall location pair may also include a respective first candidate footfall location and a respective second candidate footfall location. The method also includes clustering, by the data processing hardware, the respective first candidate footfall location into a first respective cluster group based on a height of the respective first candidate footfall location. The method further includes clustering, by the data processing hardware, the respective second candidate footfall location into a second respective cluster group based on a height of the respective second candidate footfall location. The method additionally includes generating, by the data processing hardware, a stair model by representing each of the cluster groups as a corresponding stair among a set of stairs in the robot environment and delineating each stair based on a respective midpoint between each adjacent cluster group.

[0004] Implementations of the disclosure may include one or more of the following optional features. In some examples, the method may further include for each pair of adjacent footfall locations defined by a respective first footfall location and a respective second footfall location adjacent to the respective first footfall location in position among the plurality of footfall locations, determining, by the data processing hardware, whether a vertical distance between the respective first footfall location and the respective second footfall location satisfies a stair height threshold, the stair height threshold corresponding to a height of a stair riser. In some configurations the method also includes determining, by the data processing hardware, whether a horizontal distance between the respective first footfall location and the respective second footfall location satisfies a stair depth threshold where the stair depth threshold corresponds to a depth of a stair tread. Optionally, when both (i) the vertical distance satisfies the stair height threshold and (ii) the horizontal distance satisfies the stair depth threshold, the method may include identifying, by the data processing hardware, the respective pair of adjacent footfall locations as a respective one of the plurality of candidate footfall location pairs.

[0005] In some implementations, a plurality of cluster groups indicate an orientation for the stair model, the orientation corresponding to a vector direction that a set of stairs ascend or descend within the environment. Here, after clustering each of the respective first and second footfall locations into the respective cluster groups for each candidate footfall location pair of the plurality of candidate pairs, the method may include identifying, by the data processing hardware, among the respective cluster groups, a first cluster group and a second cluster group adjacent to the first cluster group, the identified first and second clustered groups each may include one or more respective candidate footfall locations. The method further includes determining, by the data processing hardware, a respective first candidate footfall location among the one or more candidate footfall locations in the first cluster group and a respective second candidate footfall location among the one or more candidate footfall locations in the second cluster group, the respective first candidate footfall location separated by a minimum horizontal distance from the respective second candidate footfall location, and generating, by the data processing hardware, a stair edge for the stair model at a horizontal midpoint between the identified respective first candidate footfall location in the first cluster group and the identified respective second candidate footfall location in the second cluster group. Optionally, a plurality of cluster groups indicate an orientation for the stair model, the orientation corresponding to a vector direction that a set of stairs ascend or descend within the environment, and the stair edge extending in a direction perpendicular to the vector direction of the orientation for the plurality of cluster groups.

[0006] Additionally, the method may include communicating, by the data processing hardware, the stair model to a control system for the robot to navigate the stairs represented by the stair model in an autonomous drive mode. The method further includes, after generating the stair model, while the robot traverses the environment, detecting, by the data processing hardware, that the robot is approaching a location represented by the stair model and orienting, by the data processing hardware, the robot to an orientation for the stair model, the orientation corresponding to a vector direction that the set of stairs ascend or descend within the environment. In some examples, orienting the robot may include directing sensors on the robot to face the vector direction defined by the stair model. The method may include augmenting, by the data processing hardware, a perception map of the environment with the stair model. In some implementations, the robot is a quadruped robot.

[0007] Another aspect of the disclosure provides a robot configured to identify stairs from footfalls. The robot includes a body, two or more legs coupled to the body and configured to traverse an environment, and a stair modeling system in communication with the robot. The modeling system includes data processing hardware 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. The operations include receiving a plurality of footfall locations of the two or more legs traversing the environment, each respective footfall location indicating a location where one of the two of more legs of the robot contacted a support surface beneath the robot. The operations also include determining a plurality of candidate footfall location pairs based on the plurality of footfall locations, each candidate footfall location pair including a respective first candidate footfall location and a respective second candidate footfall location. For each candidate footfall location pair of the plurality of candidate footfall location pairs, the operations additionally include clustering the respective first candidate footfall location into a first respective cluster group based on a height of the respective first candidate footfall location and clustering the respective second candidate footfall location into a second respective cluster group based on a height of the respective second candidate footfall location. The operations also includes generating a stair model by representing each of the cluster groups as a corresponding stair among a set of stairs in the robot environment and delineating each stair based on a respective midpoint between each adjacent cluster group.

[0008] Implementations of the disclosure may include one or more optional features. In some implementations, for each pair of adjacent footfall locations defined by a respective first footfall location and a respective second footfall location adjacent to the respective first footfall location in position among the plurality of footfall locations, the operations include determining whether a vertical distance between the respective first footfall location and the respective second footfall location satisfies a stair height threshold, the stair height threshold corresponding to a height of a stair riser. The operations also include determining whether a horizontal distance between the respective first footfall location and the respective second footfall location satisfies a stair depth threshold, the stair depth threshold corresponding to a depth of a stair tread and when both (i) the vertical distance satisfies the stair height threshold and (ii) the horizontal distance satisfies the stair depth threshold, identifying the respective pair of adjacent footfall locations as a respective one of the plurality of candidate footfall location pairs.

[0009] In some examples, a plurality of cluster groups indicate an orientation for the stair model where the orientation corresponding to a vector direction that a set of stairs ascend or descend within the environment. In some configurations, after clustering each of the respective first and second footfall locations into the respective cluster groups for each candidate footfall location pair of the plurality of candidate pairs, the operations also include identifying among the respective cluster groups, a first cluster group and a second cluster group adjacent to the first cluster group. In these examples, the identified first and second clustered groups each may include one or more respective candidate footfall locations. Here, the operations may further include determining a respective first candidate footfall location among the one or more candidate footfall locations in the first cluster group and a respective second candidate footfall location among the one or more candidate footfall locations in the second cluster group. Additionally or alternatively, the operations may also include a respective first candidate footfall location separated by a minimum horizontal distance from the respective second candidate footfall location and generating a stair edge for the stair model at a horizontal midpoint between the identified respective first candidate footfall location in the first cluster group and the identified respective second candidate footfall location in the second cluster group.

[0010] In some configurations, a plurality of cluster groups indicate an orientation for the stair model where the orientation corresponds to a vector direction that a set of stairs ascend or descend within the environment. Here, the stair edge extends in a direction perpendicular to the vector direction of the orientation for the plurality of cluster groups. The operations further may include communicating the stair model to a control system for the robot to navigate the stairs represented by the stair model in an autonomous drive mode. The operations further may include, after generating the stair model, while the robot traverses the environment, detecting that the robot is approaching a location represented by the stair model and orienting the robot to an orientation for the stair model, the orientation corresponding to a vector direction that the set of stairs ascend or descend within the environment. Orienting the robot may include directing sensors on the robot to face the vector direction defined by the stair model. Optionally, the operations may include augmenting a perception map of the environment with the stair model. In some examples, the two or more legs may include four legs defining a quadruped robot.

[0011] 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

[0012] FIG. 1A is a perspective view of an example robot standing atop a landing of a staircase.

[0013] FIG. 1B is a schematic view of example systems of the robot of FIG. 1A.

[0014] FIG. 1C is a perspective view of an example environment traversed by the robot of FIG. 1A.

[0015] FIGS. 2A–2C are schematic views of example stair detectors of the robot of FIG. 1A.

[0016] FIG. 3 is a flow chart of an example arrangement of operations for a method of identifying stairs from footfalls.

[0017] FIG. 4 is a schematic view of an example computing device that may be used to implement the systems and methods described herein.

[0018] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0019] As legged-robots maneuver about environments, the robots may encounter terrain (e.g., human-made structures) that requires precise leg movement and foot placement (i.e., distal end placement). To provide precise leg movement and foot placement, when systems of the robot recognize different types of terrain, the movement control systems of the robot may constrain the robot's movement to traverse the terrain in order to prevent mistakes, even small mistakes, which may lead to catastrophic issues for the robot. For example, when humans traverse stairs, this task requires a degree of coordination (e.g., eye-to-foot coordination). Without the coordination, a human may misstep, slip, trip, or fall on the stairs. Robots may encounter the same misfortunes, but lack natural coordination. Therefore, robots need systems and methods to coordinate precise leg movements.

[0020] FIG. 1A is an example of an environment 10 for a robot 100. The environment 10 generally refers to a spatial area associated with some type of terrain including stairs 20, 20a-n or stair-like terrain that may be traversed by the robot 100 (e.g., using a control system 170 as shown in FIG. 1B). Systems of the robot 100 are responsible for coordinating and/or moving the robot 100 about the environment 10. As the robot 100 traverses stairs 20 or stair-like terrain and moves about the environment 10, systems of the robot 100 may analyze the terrain, plan motion trajectories for the robot 100 (e.g., with a path generator 174, a step planner 176, a body planner 178), and/or instruct the robot 100 to perform various movements (e.g., with a controller 172). The robot 100 may use various systems of the robot 100 together to attempt to successfully traverse the environment 10 while avoiding collisions C and/or damage to the robot 100 or the robot's environment 10.

[0021] Stairs 20, 20a–n generally refer to a group of more than one stair 20 (i.e., a group of n stairs 20) designed to bridge a vertical distance. To bridge the vertical distance, stairs 20a–n typically run a horizontal distance with a given rise in vertical height over a pitch (or pitch line). Each stair 20 traditionally includes a tread 22 and a riser 24. The tread 22 of a stair 20 refers to a horizontal part of the stair 20 that is stepped on while a riser 24 refers to a vertical portion of the stair 20 between each tread 22. The tread 22 of each stair 20 spans a tread depth "d" measuring from an outer edge 26 of a stair 20 to the riser 24 between stairs 20. For a residential, a commercial, or an industrial structure, some stairs 20 also include nosing as part of the edge 26 for safety purposes. Nosing, as shown in FIG. 1A, is a part of the tread 22 that protrudes over a riser 24 beneath the tread 22. For example, the nosing (shown as edge 26a) is part of the tread 22a and protrudes over the riser 24a. A set of stairs 20 may be preceded by or include a platform or support surface 12 (e.g., a level support surface). For example, a landing refers to a level platform or support surface 12 at a top of a set of stairs 20 or at a location between stairs 20. For instance, a landing occurs where a direction of the stairs 20 change or between a particular number of stairs 20 (i.e., a flight of stairs 20 that connects two floors). FIG. 1A illustrates the robot 100 standing on a landing at the top of a set of stairs 20.

[0022] Stair-like terrain more generally refers to terrain that varies in height over some distance. Stair-like terrain may resemble stairs in terms of a change in elevation (e.g., an inclined pitch with a gain in elevation or a declined pitch with a loss in elevation). However, with stair-like terrain the delineation of treads 22 and risers 24 is not as obvious. Rather, stair-like terrain may refer to terrain with tread-like portions that allow a robot to have enough traction to plant a stance limb and sequentially or simultaneously use a leading limb to ascend or to descend over an adjacent vertical obstruction (resembling a riser) within the terrain. For example, stair-like terrain my include rubble, an inclined rock scramble, damaged or deteriorating traditional stairs, etc.

[0023] Referring to FIG. 1A, the robot 100 includes a body 110 with locomotion based structures such as legs 120a–d coupled to the body 110 that enable the robot 100 to move about the environment 10. In some examples, each leg 120 is an articulable structure such that one or more joints J permit members 122 of the leg 120 to move. For instance, each leg 120 includes a hip joint J_H coupling an upper member 122, 122_U of the leg 120 to the body 110 and a knee joint J_K coupling the upper member 122_U of the leg 120 to a lower member 122_L of the leg 120. For impact detection, the hip joint J_H may be further broken down into abduction-adduction rotation of the hip joint J_H designated as "J_{Hx}" for occurring in a frontal plane of the robot 100 (i.e., a X-Z plane extending in directions of a x-direction axis A_x and the z-direction axis A_Z) and a flexion-extension rotation of the hip joint J_H designated as "J_{Hy}" for occurring in a sagittal plane of the robot 100 (i.e., a Y-Z plane extending in directions of a y-direction axis A_Y and the z-direction axis A_Z). Although FIG. 1A depicts a quadruped robot with four legs 120a–d, the robot 100 may include any number of legs or locomotive based structures (e.g., a biped or humanoid robot with two legs) that provide a means to traverse the terrain within the environment 10.

[0024] In order to traverse the terrain, each leg 120 has a distal end 124 that contacts a surface 12 of the terrain (i.e., a traction surface). In other words, the distal end 124 of the leg 120 is the end of the leg 120 used by the robot 100 to pivot, plant, or generally provide traction during movement of the robot 100. For example, the distal end 124 of a leg 120 corresponds to a foot of the robot 100. In some examples, though not shown, the distal end 124 of the leg 120 includes an ankle joint J_A such that the distal end 124 is articulable with respect to the lower member 122_L of the leg 120.

[0025] The robot 100 has a vertical gravitational axis (e.g., shown as a Z-direction axis A_Z) along a direction of gravity, and a center of mass CM, which is a point where the weighted relative position of the distributed mass of the robot 100 sums to zero. The robot 100 further has a pose P based on the CM relative to the vertical gravitational axis A_Z (i.e., the fixed reference frame with respect to gravity) to define a particular attitude or stance assumed by the robot 100. The attitude of the robot 100 can be defined by an orientation or an angular position of the robot 100 in space. Movement by the legs 120 relative to the body 110 alters the pose P of the robot 100 (i.e., the combination of the position of the CM of the robot and the attitude or orientation of the robot 100). Here, a height (i.e., vertical distance) generally refers to a distance along (e.g., parallel to) the z-direction (i.e., z-axis A_Z). The sagittal plane of the robot 100 corresponds to the Y-Z plane extending in directions of a y-direction axis A_Y and the z-direction axis A_Z. In other words, the sagittal plane bisects the robot 100 into a left and right side. Generally perpendicular to the sagittal plane, a ground plane (also referred to as a transverse plane) spans the X-Y plane by extending in directions of the x-direction axis A_X and the y-direction axis A_Y. The ground plane refers to a support surface 12 where distal ends 124 of the legs 120 of the robot 100 may generate traction to help the robot 100 move about the environment 10. Another anatomical plane of the robot 100 is the frontal plane that extends across the body 110 of the robot 100 (e.g., from a left side of the robot 100 with a first leg 120a to a right side of the robot 100 with a second leg 120b). The frontal plane spans the X-Z plane by extending in directions of the x-direction axis A_X and the z-direction axis A_Z.

[0026] When a legged-robot moves about the environment 10, the legs 120 of the robot undergo a gait cycle. Generally, a gait cycle begins when a leg 120 touches down or contacts a support surface 12 and ends when that same leg 120 once again contacts the ground surface 12. Here, touchdown is also referred to as a footfall defining a point or position where the distal end 124 of a locomotion-based structure 120 falls into contact with the support surface 12. The gait cycle may predominantly be divided into two phases, a swing phase and a stance phase. During the swing phase, a leg 120 performs (i) lift-off from the support surface 12 (also sometimes referred to as toe-off and the transition between the stance phase and swing phase), (ii) flexion at a knee joint J_K of the leg 120, (iii) extension of the knee joint J_K of the leg 120, and (iv) touchdown (or footfall) back to the support surface 12. Here, a leg 120 in the swing phase is referred to as a swing leg 120_{SW}. As the swing leg 120_{SW} proceeds through the movement of the swing phase 120_{SW}, another leg 120 performs the stance phase. The stance phase refers to a period of time where a distal end 124 (e.g., a foot) of the leg 120 is on the support surface 12. During the stance phase a leg 120 performs (i) initial support surface contact which triggers a transition from the swing phase to the stance phase, (ii) loading response where the leg 120 dampens support surface contact, (iii) mid-stance support for when the contralateral leg (i.e., the swing leg 120_{SW}) lifts-off and swings to a balanced position (about halfway through the swing phase), and (iv) terminal-stance support from when the robot's COM is over the leg 120 until the contralateral leg 120 touches down to the support surface 12. Here, a leg 120 in the stance phase is referred to as a stance leg 120_{ST}.

[0027] In order to maneuver about the environment 10, the robot 100 includes a sensor system 130 with one or more sensors 132, 132a–n (e.g., shown as a first sensor 132, 132a and a second sensor 132, 132b). The sensors 132 may include vision/image sensors, inertial sensors (e.g., an inertial measurement unit (IMU)), force sensors, and/or kinematic sensors. Some examples of sensors 132 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 configurations, the robot 100 includes two stereo cameras as sensors 132 at a front end of the body 110 of the robot 100 (i.e., a head of the robot 100 adjacent the front legs 120a–b of the robot 100) and one stereo camera as a sensor 132 at a back end of the body 110 of the robot 100 adjacent rear legs 120c–d of the robot 100. In some examples, the sensor 132 has a corresponding field(s) of view F_V defining a sensing range or region corresponding to the sensor 132. For instance, FIG. 1A depicts a field of a view F_V for the robot 100. Each sensor 132 may be pivotable and/or rotatable such that the sensor 132 may, for example, change the field of view F_V about one or more axis (e.g., an x-axis, a y-axis, or a z-axis in relation to a ground plane).

[0028] Referring to FIGS. 1A and 1B, in some implementations, the sensor system 130 includes sensor(s) 132 coupled to a joint J. In some examples, these sensors 132 couple to a motor that operates a joint J of the robot 100 (e.g., sensors 132, 132a–b). Here, these sensors 132 generate joint dynamics 134, 134_{JD} in the form of joint-based sensor data 134. Joint dynamics 134_{JD} collected as joint-based sensor data 134 may include joint angles (e.g., an upper member 122_U relative to a lower member 122_L), joint speed (e.g., joint angular velocity or joint angular acceleration), and/or joint torques experienced at a joint J (also referred to as joint forces). Here, joint-based sensor data 134 generated by one or more sensors 132 may be raw sensor data, data that is further processed to form different types of joint dynamics 134_{JD}, or some combination of both. For instance, a sensor 132 measures joint position (or a position of member(s) 122 coupled at a joint J) and systems of the robot 100 perform further processing to derive velocity and/or acceleration from the positional data. In other examples, a sensor 132 is configured to measure velocity and/or acceleration directly.

[0029] When surveying a field of view F_V with a sensor 132, the sensor system 130 generates sensor data 134 (also referred to as image data) corresponding to the field of view F_V. In some examples, the sensor data 134 is image data that corresponds to a three-dimensional volumetric point cloud generated by a three-dimensional volumetric image sensor 132. Additionally or alternatively, when the robot 100 is maneuvering about the environment 10, the sensor system 130 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, for instance, kinematic data and/or orientation data about joints J or other portions of a leg 120 of the robot 100. With the sensor data 134, a perception system 180 of the robot 100 may generate maps 182 for the terrain about the environment 10.

[0030] While the robot 100 maneuvers about the environment 10, the sensor system 130 gathers sensor data 134 relating to the terrain of the environment 10 and/or structure of the robot 100 (e.g., joint dynamics and/or odometry of the robot 100). For instance, FIG. 1A depicts the robot 100 standing on a landing (i.e., level support surface) of a set of stairs 20 as the environment 10 of the robot 100. Here, the sensor system 130 gathering sensor data 134 about the set of stairs 20. As the sensor system 130 gathers sensor data 134, a computing system 140 is configured to store, to process, and/or to communicate the sensor data 134 to various systems of the robot 100 (e.g., the control system 170, the perception system 180, an odometry system 190, and/or a stair modeler 200). In order to perform computing tasks related to the sensor data 134, the computing system 140 of the robot 100 includes data processing hardware 142 and memory hardware 144. The data processing hardware 142 is configured to execute instructions stored in the memory hardware 144 to perform computing tasks related to activities (e.g., movement and/or movement based activities) for the robot 100. Generally speaking, the computing system 140 refers to one or more locations of data processing hardware 142 and/or memory hardware 144.

[0031] With continued reference to FIGS. 1A and 1B, in some examples, the computing system 140 is a local system located on the robot 100. When located on the robot 100, the computing system 140 may be centralized (i.e., in a single location/area on the robot 100, for example, the body 110 of the robot 100), decentralized (i.e., located at various locations about the robot 100), or a hybrid combination of both (e.g., where a majority of centralized hardware and a minority of decentralized hardware). To illustrate some differences, a decentralized computing system 140 may allow processing to occur at an activity location (e.g., at motor that moves a joint of a leg 120) while a centralized computing system 140 may allow for a central processing hub that communicates to systems located at various positions on the robot 100 (e.g., communicate to the motor that moves the joint of the leg 120).

[0032] Additionally or alternatively, the computing system 140 includes computing resources that are located remotely from the robot 100. For instance, the computing system 140 may communicate via a network 150 with a remote system 160 (e.g., a remote computer/server or a cloud-based environment). Much like the computing system 140, the remote system 160 includes remote computing resources such as remote data processing hardware 162 and remote memory hardware 164. Here, sensor data 134 or other processed data (e.g., data processing locally by the computing system 140) may be stored in the remote system 160 and may be accessible to the computing system 140. In some examples, the computing system 140 is configured to utilize the remote resources 162, 164 as extensions of the computing resources 142, 144 such that resources of the computing system 140 may reside on resources of the remote system 160.

[0033] In some implementations, as shown in FIGS. 1A and 1B, the robot 100 includes a control system 170 and a perception system 180. The perception system 180 is configured to receive the sensor data 134 from the sensor system 130 and process the sensor data 134 to generate maps 182. With the maps 182 generated by the perception system 180, the perception system 180 may communicate the maps 182 to the control system 170 in order to perform controlled actions for the robot 100, such as moving the robot 100 about the environment 10. In some examples, by having the perception system 180 separate from, yet in communication with the control system 170, processing for the control system 170 may focus on controlling the robot 100 while the processing for the perception system 180 focuses on interpreting the sensor data 134 gathered by the sensor system 130. For instance, these systems 170, 180 execute their processing in parallel to ensure accurate, fluid movement of the robot 100 in an environment 10.

[0034] In some examples, the control system 170 includes at least one controller 172, a path generator 174, a step locator 176, and a body planner 178. The control system 170 may be configured to communicate with at least one sensor system 130 and any other system of the robot 100 (e.g., the perception system 180, the odometry system 190, and/or the stair modeler 200). The control system 170 performs operations and other functions using hardware 140. The controller 172 is configured to control movement of the robot 100 to traverse about the environment 10 based on input or feedback from the systems of the robot 100 (e.g., the control system 170, the perception system 180, the odometry system 190, and/or the stair modeler 200). This may include movement between poses and/or behaviors of the robot 100. For example, the controller 172 controls different footstep patterns, leg patterns, body movement patterns, or vision system sensing patterns.

[0035] In some examples, the controller 172 includes a plurality of controllers 172 where each of the controllers 172 has a fixed cadence. A fixed cadence refers to a fixed timing for a step or swing phase of a leg 120. For example, the controller 172 instructs the robot 100 to move the legs 120 (e.g., take a step) at a particular frequency (e.g., step every 250 milliseconds, 350 milliseconds, etc.). With a plurality of controllers 172 where each controller 172 has a fixed cadence, the robot 100 can experience variable timing by switching between controllers 172. In some implementations, the robot 100 continuously switches/selects fixed cadence controllers 172 (e.g., re-selects a controller 170 every 3 milliseconds) as the robot 100 traverses the environment 10.

[0036] In some implementations, the control system 170 includes specialty controllers 172 that are dedicated to a particular control purpose. For example, the control system 170 may include one or more stair controllers 172 dedicated to planning and coordinating the robot's movement to traverse a set of stairs 20. For instance, a stair controller 172 may ensure the footpath for a swing leg 120_{SW} maintains a swing height to clear a riser 24 and/or edge 26 of a stair 20. Other specialty controllers 172 may include the path generator 174, the step locator 176, and/or the body planner 178. Referring to FIG. 1B, the path generator 174 is configured to determine horizontal motion for the robot 100. For instance, the horizontal motion refers to translation (i.e., movement in the X-Y plane) and/or yaw (i.e., rotation about the Z-direction axis A_Z) of the robot 100. The path generator 174 determines obstacles within the environment 10 about the robot 100 based on the sensor data 134. The path generator 174 communicates the obstacles to the step locator 176 such that the step locator 176 may identify foot placements for legs 120 of the robot 100 (e.g., locations to place the distal ends 124 of the legs 120 of the robot 100). The step locator 176 generates the foot placements (i.e., locations where the robot 100 should step) using inputs from the perceptions system 180 (e.g., map(s) 182). The body planner 178, much like the step locator 176, receives inputs from the perceptions system 180 (e.g., map(s) 182). Generally speaking, the body planner 178 is configured to adjust dynamics of the body 110 of the robot 100 (e.g., rotation, such as pitch or yaw and/or height of COM) to successfully move about the environment 10.

[0037] The perception system 180 is a system of the robot 100 that helps the robot 100 to move more precisely in a terrain with various obstacles. As the sensors 132 collect sensor data 134 for the space about the robot 100 (i.e., the robot's environment 10), the perception system 180 uses the sensor data 134 to form one or more maps 182 for the environment 10. Once the perception system 180 generates a map 182, the perception system 180 is also configured to add information to the map 182 (e.g., by projecting sensor data 134 on a preexisting map) and/or to remove information from the map 182.

[0038] Referring further to FIG. 1B, the odometry system 190 is configured to measure where the robot 100 is located within a world reference frame (e.g., the environment 10) and how fast the robot 100 is moving in that world reference frame. In other words, the odometry system 190 generates odometry information 192 as one or more estimations (e.g., measurements) for a characteristic of the robot 100 relative to a world reference frame. In some examples, the odometry system 190 receives sensor data 134 from a sensor 132 such as an IMU (e.g., accelerometer(s) and/or gyro(s)). With the sensor data 134, the odometry system 190 may generate odometry information 192 based on an assumption that when a distal end 124 of a leg 120 is in contact with the ground surface 12 and not slipping, the distal end 124 is stationary. By combining this assumption with the sensor data 134, the odometry system 190 generates odometry information 192 regarding robot motion relative to the world reference frame (e.g., the environment 10). In other words, the odometry system 190 accounts for kinematics and inertial measurements to produce estimations about the robot 100 with respect to the world reference frame.

[0039] In some configurations, the robot 100 is configured to traverse the environment 10 autonomously. For example, the robot 100 has an autonomous mode that, when engaged (e.g., by an operator of the robot 100), allows the system(s) of the robot 100 to operate the robot 100 to move about the environment 10 and/or perform actions within the environment 10 without further input from an external source (e.g., an entity that operates or supervisors the robot 100 by providing inputs to the robot 100). In order to engage the autonomous mode, the robot 100 first surveys the environment 10 to generate one or more maps 182 using the perception system 180. In other words, prior to the autonomous mode, the robot 100 undertakes a mapping process to collect sensor data 134 of the environment 10 that will be autonomously or semi-autonomously traversed. In some examples, an operator manually drives the robot 100 (i.e., moves the robot 100 by user input) through the environment 10 (e.g., using a remote controller 172). By the robot 100 first gathering its surroundings in the environment 10, the mapping process provides environmental context to systems of the robot 100 to enable the systems to autonomously operate the robot 100.

[0040] Generally speaking, the mapping process calibrates the robot 100 to features in the environment 10; allowing the robot 100 to have the ability to autonomously or semi-autonomously operate subsequent to the mapping process. Here, semi-autonomously refers to the ability of the robot 100 to perform certain tasks (e.g., specialized tasks) independent of external inputs. For example, the robot 100 has a stair mode where the robot 100 is able to traverse stairs without external inputs or a palletizing mode where the robot 100 packs or unpacks boxes in an independent manner. In some configurations, an operator of the robot 100 identifies (e.g., turns on a mode of the robot 100) that he or she wishes to operate the robot 100 autonomously or semi-autonomously (e.g., for specialized autonomous activities). Once the robot 100 receives such an identification, systems associated with the robot 100 may prompt the operator to perform the initial mapping process if it has not been previously performed by the robot 100.

[0041] During the initial mapping process, there are particular features in the environment 10 that the robot 100 is trying to recognize. One of these features is stairs 20. Particularly, stairs 20 are a feature that may affect the robot's navigation of an environment 10. For instance, stairs 20 may pose a sudden hazard to the robot 100 when the robot 100 encounters stairs 20. If the robot 100 approaches the stairs 20 from above and did not know ahead of time that the stairs 20 existed, the robot 100 may not have much time to decide whether the sudden perceived drop-off in elevation is actually safe for the robot 100 to navigate. For example, the robot 100 while navigating a hallway suddenly approaches the end of the hallway and first perceives that stairs 20 ascend/descend from a doorway perpendicular to the end of the hallway. With sensors 132 unable to see through walls, the walls of the hallway would make the stairs 20 directly off the hallway seem invisible to the robot 100 until the robot 100 was located in front of the stairs 20. Depending to the gait of the robot 100 and/or speed of approach, sudden decision making by the robot 100 to navigate the stairs 20 may be problematic and/or dangerous for the robot 100 and/or its surroundings.

[0042] If instead, a robot 100 knew that it was approaching stairs 20, the robot 100 may prepare to navigate the stairs 20; increasing the robot's ability to navigate the stairs 20 successfully. In other words, the robot 100 may position its body 110, legs 120, or structure to improve navigation of the stairs 20. For instance, when the robot 100 approaches stairs 20, it may change the angles or heights of its sensors 132 to increase its capabilities to perceive the stairs 20 (e.g., avoid potential occlusions). The robot 100, with its sensors 132, may peer upwards, downwards, and/or change its body height to optimize its perception of the stair structure. In some examples, the robot 100 is configured to navigate the stairs 20 at a particular orientation or pose P (e.g., alignment) such that the robot 100 minimizes structural collisions with the stairs 20 themselves. As an example, the robot 100 descends stairs backwards (e.g., head up and rear legs 120 first) to prevent articulation of its locomotion structures 120 from colliding with the risers 24 of the stairs 20. Here, to descend the stairs 20 backwards, the robot 100 may need to turn around. In some implementations, the robot 100 may be configured to center itself with respect to the stairs 20 in order to provide the robot 100 with the greatest lateral space as the robot 100 ascends/descends stairs 20. Each of these particular alignments, used to improve the ability of the robot 100 to navigate the stairs 20, may be planned in advance when the initial mapping process identifies stairs 20 in the environment 10.

[0043] In some configurations, the robot 100 is able to automatically engage a stair mode for the robot 100. For example, when the robot 100 knows at least a rough location of stairs 20 within the environment 10, the robot 100 activates the stair mode when the robot 100 is adjacent to or within a threshold distance of the stairs 20. In some implementations, the robot 100 is configured to prevent or to warn a user or operator from engaging the stair mode when, based on the initial mapping process, the robot 100 is aware that the robot 100 is not within a particular range of stairs 20. By selectively activating the stair mode only when stairs 20 are imminent (e.g., within the threshold distance/range), the robot 100 may conserve processing resources (e.g., CPU usage) and/or intelligently use computing resources. For example, the robot 100 is aware that the robot 100 does not need to detect stairs 20 and optimizes its detection for other features in the environment 10.

[0044] With reference to FIGS. 1A–1C, as the robot 100 undergoes the initial mapping process, the robot 100 establishes waypoints Wp, $Wp_{1\text{-}i}$ in the environment 10. These waypoints may incorporate odometry information from the odemetry system 190 along with information about the robot 100 such as pose P and other kinematics of the robot 100. When the robot 100 traverses the environment 10 during the initial mapping process, the robot 100 establishes waypoints Wp periodically (e.g., every couple meters or when the robot 100 changes direction). An edge E connects each waypoint Wp to its neighboring waypoints Wp along the traversal path taken by the robot 100. Each edge E may serve as storage for information about the robot 100 that occurred while the robot 100 travels along the edge E between two waypoints Wp. For instance, the edge E stores footfalls 128 that occurred by the robot 100 when the robot 100 moved between the waypoints Wp along the edge E.

[0045] A footfall 128, much like a footprint, refers to a spatial location where a distal end 124 of a locomotion structure 120 of the robot 100 contacted the support surface 12. A footfall 128 may also be referred to interchangeably as a footfall location. In some examples, the footfall 128 corresponds to a touchdown for a foot 124 of a leg 120 of the robot 100. Since a footfall 128 includes a spatial location in the environment 10 where a touchdown occurred, the footfall 128 includes coordinate/odometry information to identify the location in the environment 10. The location information corresponding to a footfall 128 may be relative location information (e.g., relative to a position of a waypoint Wp or other feature in the environment 10) or global location information. For example, the footfall 128 has a three dimensional coordinate position relative to the global world reference frame (e.g., corresponds to an x, y, z location).

[0046] As footfalls 128 occur while the robot 100 traverses the environment 10 during the initial mapping process, systems (e.g., the sensor system 130, perception system 180, and/or odometry system 190) of the robot 100 are able to determine that a footfall 128 occurs and to store the footfall 128 (e.g., locational/contact information about the footfall 128 and/or the robot 100 at the time of the footfall 128). In some examples, each time a foot 124 of the robot 100 contacts a support surface 12, one or more sensors 132 of the robot 100 detect the contact as a footfall 128. In some implementations, systems (e.g., the perception system 180 and/or odometry system 190) of the robot 100 record footfalls 128 by querying sensor data 134 from contact detection sensors 132 (e.g., sensors 132 that measure forces experienced at the legs/feet 120, 124 of the robot 100) to determine whether a particular foot 124 of the robot 100 is in contact with a support surface 12 at the time of the query. When a sensor 132 indicates that a particular foot 124 is indeed in contact with a support surface 12, the systems of the robot 100 determine the location of the contact (e.g., in the world reference frame) and store this information along with the footfall 128. For example, the perception system 180 stores footfalls 128 in data structures associated with edges E for a map 182 generated during the initial mapping process.

[0047] FIG. 1C illustrates an initial map 182 of the environment 10 generated by the perception system 180 during execution of the initial mapping process. Here, the robot 100 was driven in a loop that ascended two sets of stairs 20 and descended two sets of stairs 20. As the robot 100 moved through loop within the environment 10 during the initial mapping process, systems of the robot 100 determine footfalls 128 that occur. In this example, the sets of footfalls 128, 128a–n are associated (e.g., stored in data structures) with edges E, E_{1-n} (e.g., shown as ten edges E_{1-10}) between the waypoints Wp, Wp_{1-14}, but the footfalls 128 may be stored in any data location accessible to systems of the robot 100. Based on these footfalls 128, a stair detector 200 is configured to determine whether a set of stairs 20 exists based on the footfalls 128. Here, the stair detector 200 identifies four stair models 202, 202a–d along the path traveled by the robot 100. In other words, a stair detector 200 functions at or in conjunction with the initial mapping process to receive footfalls 128 as inputs and generate one or more stair models

202 as outputs. Additionally or alternatively, the stair detector 200 may be configured to first determine footfalls 128 from sensor data 134 and then generate one or more stair models 202 when the footfalls 128 indicate a strong likelihood that stairs 20 exist within the environment 10. The stair detector 200 may be part of other systems of the robot 100 (e.g., the perception system 180) or its own dedicated system of the robot 100 (e.g., includes its own dedicated processing resources).

[0048] Referring to FIGS. 2A–2C, the stair detector 200 includes a candidate identifier 210 and a stair recognizer 220. The candidate identifier 210 is configured to determine a pair of footfalls 128 defined as a candidate pair 212 of footfalls 128. A candidate pair 212 refers to two footfalls 128 nearby in position that have a spatial relationship indicative of a structure of a stair 20. In other words, the identifier 210 is configured to identify two footfalls 128 whose vertical and horizontal spacing appears to correspond to a foot 124 of the robot 100 moving from a first stair 20a to a second stair 20b to clear a riser 24. Because in reality the robot 100 moves its feet 124 from a tread 22a of first stair 20a to a tread 22b of a second stair 20b (either above or below the first stair 20a) over a riser 24, this movement pattern will be reflected in the pattern of footfalls 128. Therefore, the identifier 210 is configured with parameters set to correspond to threshold distances (e.g., a thresholds for a vertical distance and a horizontal distance) that are typical of a stair structure and/or movement patterns for the legs 120 of the robot 100.

[0049] In some implementations, the parameters include a stair height threshold 214 and a stair depth threshold 216. When identifying a candidate pair 212 of footfalls 128, the identifier 210 determines whether the footfalls 128 satisfy each threshold 214, 216 before classifying the footfalls 128 as a candidate pair 212. For instance, The identifier 210 may determine whether the footfalls 128 satisfy these thresholds 214, 216 in either order (e.g., first the stair height threshold 214 and then the stair depth threshold 216 or vice versa). In order to determine whether the two footfalls 128 satisfy these thresholds 214, 216, the identifier 210 determines a locational position of each footfall 128 (e.g., x, y, z coordinate position) and determines a distance between each footfall 128 (e.g., with respect to each coordinate - Δx, Δy, Δz). Here, a height (e.g., vertical distance) generally refers to a measurement (e.g., Δz) in the z-direction along an axis parallel to a gravitational axis of the robot 100 while the depth (e.g., horizontal distance) refers to a measurement in the XY plane (e.g., Δx or Δy) that occurs perpendicular to the gravitation axis of the robot 100. With the distance between each footfall 128 of a potential candidate pair, the identifier 210 compares these distances to the appropriate thresholds 214, 216. For instance, the identifier 210 compares the height distance measurement between the footfalls 128 to the stair height threshold 214 and a depth distance measurement between the footfalls 128 to the stair depth threshold 216.

[0050] In some configurations, the stair height threshold 214 corresponds to a height range between a particular stair height minimum 204_{min} and a stair height maximum 204_{max}. In other words, based on the gait of robot 100 and/or the structural configuration of the robot 100, there is a particular height for the robot 100 where, above this height, the robot 100 needs to change its natural gait/swing trajectory to ensure clearance of the this height change for the foot 124 of the robot 100 while, below this height, the robot 100 does not need to alter its natural gait/swing trajectory. Due to this inherent swing trajectory, the stair detector 200 may be configured to ignore modeling a stair 20 when the height change is lower than the height minimum 204_{min}. That is, although shallow stairs 20 with riser heights less than the height minimum 204_{min} exist, these stairs 20 do not pose much of a navigability risk to the robot 100 and/or need special gait instructions.

[0051] Similarly, due to range of motion limitations and/or gait limitations, there is a particular height above which the robot 100 cannot step without a more powerful movement (e.g., a jump). In other words, the robot 100 has a maximum swing height that ensures its feet 124 clear an object below this height. As such, a height greater than this height (i.e., the stair height maximum 204_{max}), even if the height of a stair riser 24 (e.g., a large amphitheater stair), is an outlier height that stair detector 200 may be configured to ignore when modeling a stair 20. With the stair height minimum 204_{min} and the stair height maximum 204_{max}, the identifier 210 may configure the stair height threshold 214 as a range of height values between the stair height minimum 204_{min} and the stair height maximum 204_{max}. In other words, if the height measurement between two footfalls 128 is a between the stair height minimum 204_{min} and the stair height maximum 204_{max}, the height measurement satisfies the stair height threshold 214. In contrast, when the height measurement is greater than the stair height maximum 204_{max} or less than the stair height minimum 204_{min}, the height measurement fails to satisfy the stair height threshold 214.

[0052] Referring to FIG. 2B, when the identifier 210 evaluates a potential candidate pair of footfalls 128, the identifier 210, in a general sense, determines whether footfalls 128 (e.g., shown as a first footfall 128, 128a and a second footfall 128, 128b) near each other in location satisfy the thresholds 214, 216 while also determining whether another footfall 128 does not invalidate the pair of footfalls 128, 128a–b as a candidate pair 212. For example, although two footfalls 128 may be near each other, a third footfall 128 may occur between the two footfalls 128. For instance, FIG. 2B illustrates, in the upper right-hand corner, the identifier 210 evaluating a potential candidate pair 128a–b with a dotted line segment between two footfalls 128 (e.g., shown as grey shaded ovals). Here, no intervening footfall 128 exists between these two footfalls 128a–b. Yet in the other depicted examples (e.g., labeled 1–6), such as a third example, an intervening footfall 128 exists between the two footfalls 128 being evaluated by the identifier 210; invalidating the pair of footfalls 128 as a candidate pair 212. In some examples, the identifier 210 tries to best alleviate the issue of an intervening invalidating footfalls 128 between a potential candidate pair by performing the candidate pair determination on footfalls 128 that are most adjacent to each other (e.g., based on coordinate position). In FIG. 2B, the examples outlined with a darker box refer to footfalls 128 that the identifier 210 has determined to be candidate pairs 212, while the examples with a dotted box outline potential candidate pairs that the identifier 210 determined to not be a candidate pair 212. Here, the second example fails to satisfy the stair height threshold 214. The third example has an intervening footfall 128. The sixth example is an evaluation of the third example with respect to adjacent footfalls 128, but the potential candidate pair does not satisfy either threshold 214, 216 (and is below the stair height minimum 204_{min}).

[0053] The stair detector 200 also includes a stair recognizer 220. The stair recognizer 220, is configured to, based on candidate pairs 212 of footfalls 128, determine stairs 20 corresponding to the footfalls 128 of the candidate pairs 212. Stated differently, the identifier 210 of the detector 200 is tasked with identifying footfalls 128 that occur on treads 22 of a stair 20 based positional data while the recognizer 220 is then configured to model each stair 20 for a stair model 202 based on clusters of footfalls 128 corresponding to candidate pairs 212. With this approach, the identifier 210 serves as a form of a filter that filters out footfalls 128 that likely do not exist on stairs 20 and the recognizer 220 constructs the model 202 based on the remaining filtered footfall data.

[0054] Referring to FIG. 2C, the stair recognizer 220 is configured to identify that each footfall 128 of a candidate pair 212 corresponds to an adjacent stair 20 on a set of stairs 20. In other words, a first footfall 128 of a candidate pair 212 occurs on a first stair 20a while a second footfall 128 of the candidate pair 212 occurs on a second stair 20b (e.g., above or below the first stair 20a). With this information, the recognizer 220 is configured to cluster footfalls 128 of the candidate pairs 212 communicated from the identifier 210. In some examples, the recognizer 220 clusters the footfalls 128 based on a height (e.g., z-coordinate) corresponding to each footfall 128. For example, the recognizer 220 clusters each footfall 128 of the candidate pairs 212 that is within a particular height tolerance of each other. In some examples, the recognizer 220 determines average height intervals corresponding to all footfalls 128 of candidate pairs 212. For instance, in a three stair 20a–c example, the recognizer 220 identifies three bands of heights (e.g., three discrete height range intervals) within all the footfalls 128 of the candidate pairs 212; a first band that corresponds to a first stair 20a, a second band that corresponds to a second stair 20b, and a third band that corresponds to a third stair 20c. From this identification, the recognizer 220 defines each band as a cluster group 222 corresponding to a stair 20 for the model 202. Here, the recognizer 220 generates a stair 20 for the model 202 for each cluster group 222. When recognizer 220 generates a stair 20 for the model 202 from each cluster group 222, the recognizer 220 may define the stair 20 in the model 202 to exist at a z-coordinate height corresponding to the average height for all footfalls 128 within a given cluster group 222. By defining a stair 20 of the model 202 at the average height for the cluster group 222, the recognizer 220 helps ensure that some flexion in the stair structure from a foot contact during footfall generation does not lead to inaccuracies for the actual z-height of a top surface of a tread 22 of a stair 20.

[0055] Once the footfalls 128 from the candidate pairs 212 are defined by cluster groups 222, the recognizer 220 has established where it believes top surfaces for treads 22 of the stairs 20 to be, but the stair model 202 still lacks some form of horizontal delineation between each stair 20 in the model 202. In some examples, the recognizer 220 defines an edge 26 of each stair 20 to be where one cluster group 222 changes to its neighboring cluster group 222. For instance, FIG. 2C depicts a first line 224, 224a as a stair edge 26 for the model 202 between the first cluster group 222, 222a and a second cluster group 222b. In some implementations, the recognizer 220 defines the position of the line 224 by a midpoint between cluster groups 222. Here, the recognizer 220 identifies a footfall 128 in a cluster group 222 that is most adjacent to a footfall 128 in a second, neighboring cluster group 222. For instance, FIG. 2C illustrates the two most adjacent footfalls 128 in a darker outline than the rest in the first cluster group 222a and the second cluster group 222b. Although these two footfalls 128 may correspond to different candidate pairs 212, the footfall 128 in the first cluster group 222a is the closest (e.g., in XY position) to the footfall 128 in the second cluster group 222b. As such, the recognizer 220 defines the line based on the midpoint (e.g., shown as midpoint MP) between these two footfalls 128. For example, the first line 224a corresponding to an edge 26 of the second stair 20b in the model 202 passes through the midpoint MP.

[0056] In some configurations, the model 202 designates an orientation for the stairs 20 of the model 202. In other words, the model 202 defines a vector direction V_D for the stairs 20 of the model 202. Here, the vector direction V_D refers to a vector that defines the direction and/or slope of the stairs 20 in three-dimensional space. When the robot 100 uses the model 202 navigating the environment 10, the robot 100 may orient itself or its sensors 132 with the vector direction V_D to aid the robot 100 in perceiving the stairs 20 or aligning with the stairs 20. In some examples, the robot 100 navigates the stairs 20 better when the body 110 of the robot 100 is aligned with the vector direction V_D than not. This alignment may afford the robot 100 the greatest lateral deviation while traversing the stairs 20 (e.g., by traversing the center of the stairs 20) and potentially prevent the robot 100 from colliding with the stair structure itself. In some configurations, in order to determine the vector direction V_D, the recognizer 220 determines the center of each cluster group 222 (e.g., the centroids of the collection of footfalls 128 included in a cluster group 222). Here, the recognizer 220 define the vector direction V_D as a vector extending from a cluster group 222 at a first end of a stair model 202 to a cluster group 222 at a second end of the stair model 202. FIG. 2C, therefore, depicts the vector direction V_D extending through center points of the first, second, and third cluster groups 222a–c.

[0057] FIG. 3 is a flowchart of an example arrangement of operations for a method of identifying stairs 20 from footfalls 128. At operations 302, the method 300 receives a plurality of footfall locations 128 indicating a location where a leg 120 of the robot 100 contacted a support surface 12 beneath the robot 100. At operation 304, the method 300 determines a plurality of candidate footfalls location pairs 212 based on the plurality of footfall locations 128. Here, each candidate footfall location pair 212 includes a respective first candidate footfall location 212, 212a and a second respective candidate footfalls location 212, 212b. Operation 306 includes a first operation 306a and a second operation 306b for each candidate footfall location pair 212 of the plurality of candidate pairs 212. At operations 306a, the method 300 clusters the respective first candidate footfall location 128, 128a into a first respective cluster group 222, 222a based on a height of the respective first candidate footfall location 128, 128a. At operation 306b, the method 300 clusters the respective second candidate footfall location 128, 128b into a second respective cluster group 222, 222b based on a height of the respective second candidate footfall location 128, 128b. At operation 308, the method 300 generates a stair model 202 by representing each of the cluster groups 222 as a corresponding stair 20 among a set of stairs 20, 20a–n in the robot environment 10 and delineating each stair 20 based on a respective midpoint MP between each adjacent cluster group 222.

[0058] FIG. 4 is schematic view of an example computing device 400 that may be used to implement the systems and methods described in this document. The computing device 400 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.

[0059] The computing device 400 includes a processor 410 (e.g., data processing hardware 142, 162), memory 420 (e.g., memory hardware 144, 164), a storage device 430, a high-speed interface/controller 440 connecting to the memory 420 and high-speed expansion ports 450, and a low speed interface/controller 460 connecting to a low speed bus 470 and a storage device 430. Each of the components 410, 420, 430, 440, 450, and 460, are interconnected using various busses, and may be mounted on a common motherboard or in other manners as appropriate. The processor 410 can process instructions for execution within the computing device 400, including instructions stored in the memory 420 or on the storage device 430 to display graphical information for a graphical user interface (GUI) on an external input/output device, such as display 480 coupled to high speed interface 440. 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 400 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).

[0060] The memory 420 stores information non-transitorily within the computing device 400. The memory 420 may be a computer-readable medium, a volatile memory unit(s), or non-volatile memory unit(s). The non-transitory memory 420 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 400. 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.

[0061] The storage device 430 is capable of providing mass storage for the computing device 400. In some implementations, the storage device 430 is a computer-readable medium. In various different implementations, the storage device 430 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 420, the storage device 430, or memory on processor 410.

[0062] The high speed controller 440 manages bandwidth-intensive operations for the computing device 400, while the low speed controller 460 manages lower bandwidth-intensive operations. Such allocation of duties is exemplary only. In some implementations, the high-speed controller 440 is coupled to the memory 420, the display 480 (e.g., through a graphics processor or accelerator), and to the high-speed expansion ports 450, which may accept various expansion cards (not shown). In some implementations, the low-speed controller 460 is coupled to the storage device 430 and a low-speed expansion port 490. The low-speed expansion port 490, 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.

[0063] The computing device 400 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a standard server 400a or multiple times in a group of such servers 400a, as a laptop computer 400b, as part of a rack server system 400c, or as part of the robot 100.

[0064] 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.

[0065] 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.

[0066] 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.

[0067] 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.

[0068] 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 (300) comprising:

receiving, at data processing hardware (410), a plurality of footfall locations (128) of a robot (100) traversing an environment (10), each respective footfall location (128) indicating a location where a leg (120) of the robot (100) contacted a support surface (12) beneath the robot (100);

determining, by the data processing hardware (410), a plurality of candidate footfall location pairs (212) based on the plurality of footfall locations (128), each candidate footfall location pair (212) comprising a respective first candidate footfall location (212) and a respective second candidate footfall location (212);

for each candidate footfall location pair (212) of the plurality of candidate footfall location pairs (212):

clustering, by the data processing hardware (410), the respective first candidate footfall location (212) into a first respective cluster group (222) based on a height of the respective first candidate footfall location (212); and

clustering, by the data processing hardware (410), the respective second candidate footfall location (212) into a second respective cluster group (222) based on a height of the respective second candidate footfall location (212); and

generating, by the data processing hardware (410), a stair model (202) by representing each of the cluster groups (222) as a corresponding stair (20) among a set of stairs (20) in the robot environment (10) and delineating each stair (20) based on a respective midpoint (MP) between each adjacent cluster group (222).

2. The method (300) of claim 1, further comprising:

for each pair of adjacent footfall locations (212) defined by a respective first footfall location (128) and a respective second footfall location (128) adjacent to the respective first footfall location (128) in position among the plurality of footfall locations (128):

determining, by the data processing hardware (410), whether a vertical distance between the respective first footfall location (128) and the respective second footfall location (128) satisfies a stair height threshold (214), the stair height threshold (214) corresponding to a height of a stair riser (24);

determining, by the data processing hardware (410), whether a horizontal distance between the respective first footfall location (128) and the respective second footfall location (128) satisfies a stair depth threshold (216), the stair depth threshold (216) corresponding to a depth of a stair tread (22); and

when both (i) the vertical distance satisfies the stair height threshold (214) and (ii) the horizontal distance satisfies the stair depth threshold (216), identifying, by the data processing hardware (410), the respective pair of adjacent footfall locations (128) as a respective one of the plurality of candidate footfall location pairs (212).

3. The method (300) of any of claims 1 or 2, wherein a plurality of cluster groups (222) indicate an orientation for the stair model (202), the orientation corresponding to a vector direction (V_D) that a set of stairs (20) ascend or descend within the environment (10).

4. The method (300) of any of claims 1–3, further comprising, after clustering each of the respective first and second footfall locations (128) into the respective cluster groups (222) for each candidate footfall location pair (212) of the plurality of candidate pairs (212):

identifying, by the data processing hardware (410), among the respective cluster groups (222), a first cluster group (222) and a second cluster group (222) adjacent to the first cluster group (222), the identified first and second clustered groups (222) each comprising one or more respective candidate footfall locations (128);

determining, by the data processing hardware (410), a respective first candidate footfall location (212) among the one or more candidate footfall locations (212) in the first cluster group (222) and a respective second candidate footfall location (212) among the one or more candidate footfall locations (212) in the second cluster group (222), the respective first candidate footfall location (212) separated by a minimum horizontal distance from the respective second candidate footfall location (212); and generating, by the data processing hardware (410), a stair edge (26) for the stair model (202) at a horizontal midpoint (MP) between the identified respective first candidate footfall location (212) in the first cluster group (222) and the identified respective second candidate footfall location (212) in the second cluster group (222).

5. The method (300) of claim 4, wherein:

a plurality of cluster groups (222) indicate an orientation for the stair model (202), the orientation corresponding to a vector direction (V_D) that a set of stairs (20) ascend or descend within the environment (10), and

the stair edge (26) extending in a direction perpendicular to the vector direction (V_D) of the orientation for the plurality of cluster groups (222).

6. The method (300) of any of claims 1–5, further comprising communicating, by the data processing hardware (410), the stair model (202) to a control system (170) for the robot (100) to navigate the stairs (20) represented by the stair model (202) in an autonomous drive mode.

7. The method (300) of any of claims 1–6, further comprising, after generating the stair model (202), while the robot (100) traverses the environment (10):

detecting, by the data processing hardware (410), that the robot (100) is approaching a location represented by the stair model (202); and

orienting, by the data processing hardware (410), the robot (100) to an orientation for the stair model (202), the orientation corresponding to a vector direction (V_D) that the set of stairs (20) ascend or descend within the environment (10).

8. The method (300) of claim 7, wherein orienting the robot (100) comprises directing sensors (132) on the robot (100) to face the vector direction (V_D) defined by the stair model (202).

9. The method (300) of any of claims 1–8, further comprising augmenting, by the data processing hardware (410), a perception map (182) of the environment (10) with the stair model (202).

10. The method (300) of any of claims 1–9, wherein the robot (100) is a quadruped robot (100).

11. A robot (100) comprising:

a body (110);

two or more legs (120) coupled to the body (110) and configured to traverse an environment (10);

a stair modeling system (200) in communication with the robot (100), the modeling system (200) comprising data processing hardware (410) and memory hardware (420) in communication with the data processing hardware (410), the memory hardware (420) storing instructions that when executed on the data processing hardware (410) cause the data processing hardware (410) to perform operations comprising:

receiving a plurality of footfall locations (128) of the two or more legs (120) traversing the environment, each respective footfall location (128) indicating a location where one of the two of more legs (120) of the robot (100) contacted a support surface beneath the robot (100);

determining a plurality of candidate footfall location pairs (212) based on the plurality of footfall locations (128), each candidate footfall location pair (212) comprising a respective first candidate footfall location (212) and a respective second candidate footfall location (212);

for each candidate footfall location pair (212) of the plurality of candidate footfall location pairs (212):

clustering the respective first candidate footfall location (212) into a first respective cluster group (222) based on a height of the respective first candidate footfall location (212); and clustering the respective second candidate footfall location (212) into a second respective cluster group (222) based on a height of the respective second candidate footfall location (212); and

generating a stair model (202) by representing each of the cluster groups (222) as a corresponding stair (20) among a set of stairs (20) in the robot environment (10) and delineating each stair (20) based on a respective midpoint (MP) between each adjacent cluster group (222).

12. The robot (100) of claim 11, wherein the operations further comprise:

for each pair of adjacent footfall locations (212) defined by a respective first footfall location (128) and a respective second footfall location (128) adjacent to the respective first footfall location (128) in position among the plurality of footfall locations (128):

determining whether a vertical distance between the respective first footfall location (128) and the respective second footfall location (128) satisfies a stair height threshold (214), the stair height threshold (214) corresponding to a height of a stair riser (24);

determining whether a horizontal distance between the respective first footfall location (128) and the respective second footfall location (128) satisfies a stair depth threshold (216), the stair depth threshold (216) corresponding to a depth of a stair tread (22); and

when both (i) the vertical distance satisfies the stair height threshold (214) and (ii) the horizontal distance satisfies the stair depth threshold (216), identifying the respective pair of adjacent footfall locations (128) as a respective one of the plurality of candidate footfall location pairs (212).

13. The robot (100) of any of claims 11 or 12, wherein a plurality of cluster groups (222) indicate an orientation for the stair model (202), the orientation corresponding to a vector direction (V_D) that a set of stairs (20) ascend or descend within the environment (10).

14. The robot (100) of any of claims 11–13, wherein the operations further comprise, after clustering each of the respective first and second footfall locations (128) into the respective cluster groups (222) for each candidate footfall location pair (212) of the plurality of candidate pairs (212):

identifying among the respective cluster groups (222), a first cluster group (222) and a second cluster group (222) adjacent to the first cluster group (222), the identified first and second clustered groups (222) each comprising one or more respective candidate footfall locations (128);

determining a respective first candidate footfall location (212) among the one or more candidate footfall locations (212) in the first cluster group (222) and a respective second candidate footfall location (212) among the one or more candidate footfall locations (212) in the second cluster group (222), the respective first candidate footfall location (212) separated by a minimum horizontal distance from the respective second candidate footfall location (212); and

generating a stair edge (26) for the stair model (202) at a horizontal midpoint (MP) between the identified respective first candidate footfall location (212) in the first cluster group (222) and the identified respective second candidate footfall location (212) in the second cluster group (222).

15. The robot (100) of claim 14, wherein:

a plurality of cluster groups (222) indicate an orientation for the stair model (202), the orientation corresponding to a vector direction (V_D) that a set of stairs (20) ascend or descend within the environment (10), and

the stair edge (26) extending in a direction perpendicular to the vector direction (V_D) of the orientation for the plurality of cluster groups (222).

16. The robot (100) of any of claims 11–15, wherein the operations further comprise communicating the stair model (202) to a control system (170) for the robot (100) to

navigate the stairs (20) represented by the stair model (202) in an autonomous drive mode.

17. The robot (100) of any of claims 11–16, wherein the operations further comprise, after generating the stair model (202), while the robot (100) traverses the environment (10):

detecting that the robot (100) is approaching a location represented by the stair model (202); and

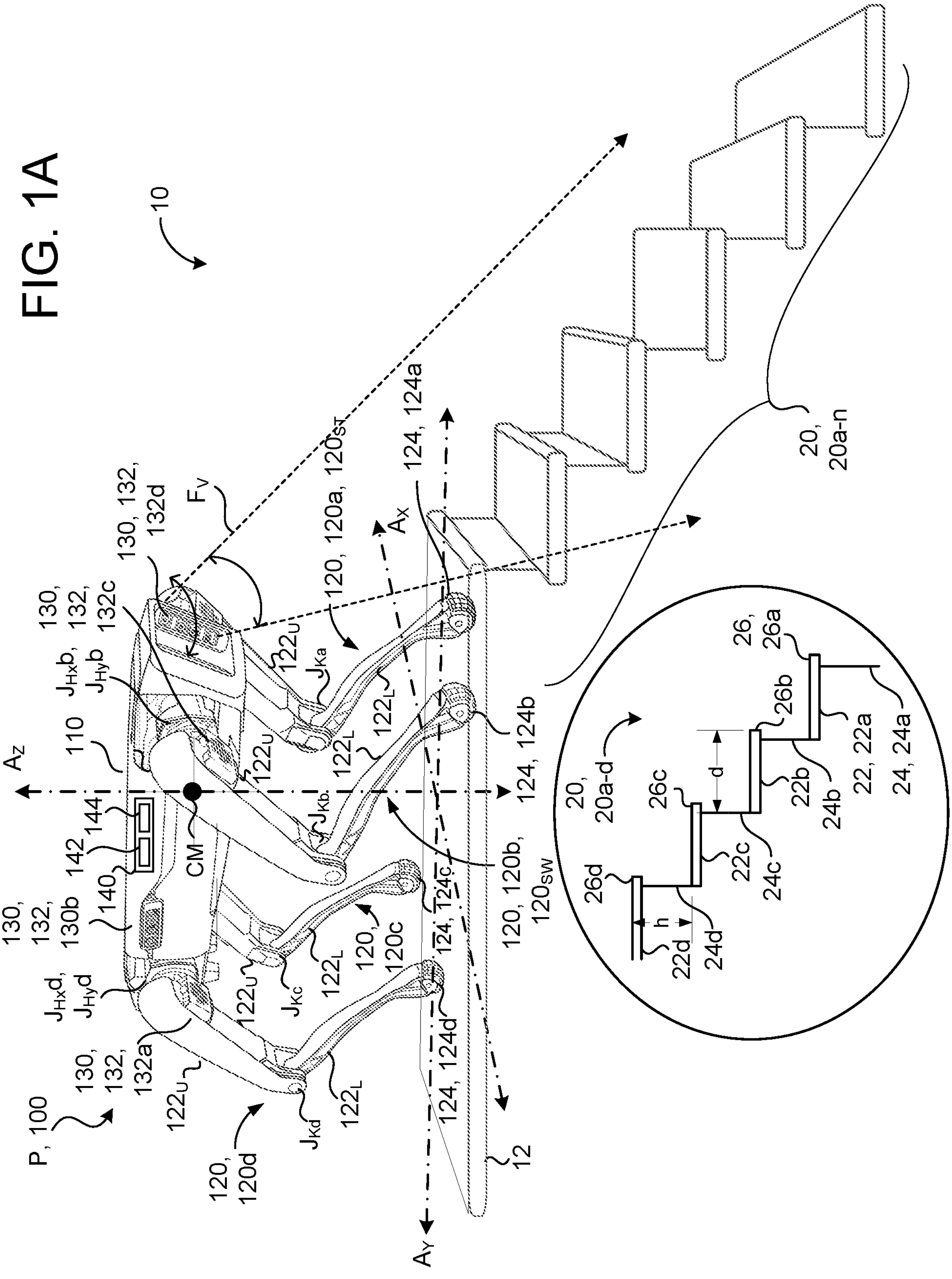
orienting the robot (100) to an orientation for the stair model (202), the orientation corresponding to a vector direction (V_D) that the set of stairs (20) ascend or descend within the environment (10).

18. The robot (100) of claim 17, wherein orienting the robot (100) comprises directing sensors (132) on the robot (100) to face the vector direction (V_D) defined by the stair model (202).

19. The robot (100) of any of claims 11–18, wherein the operations further comprise a perception map (182) of the environment (10) with the stair model (202).

20. The robot (100) of any of claims 11–19, wherein the two or more legs (120) comprise four legs (120) defining a quadruped robot (100).

FIG. 1A


10
P, 100
130, 132, 132a
130, 132, 130b
130, 132, 132c
130, 132, 132d
$J_{Hx}d$, $J_{Hy}d$
$J_{Hx}b$, $J_{Hy}b$
140
142
144
110
CM
A_Z
A_X
A_Y
F_V
122_U
122_L
J_{Ka}
J_{Kb}
J_{Kc}
J_{Kd}
120, 120a, 120_{ST}
120, 120b, 120_{SW}
120, 120c
120, 120d
124, 124a
124, 124b
124, 124c
124, 124d
12
20, 20a-n
20, 20a-d
26, 26a
26b
26c
26d
22, 22a
22b
22c
22d
24, 24a
24b
24c
24d
d
h

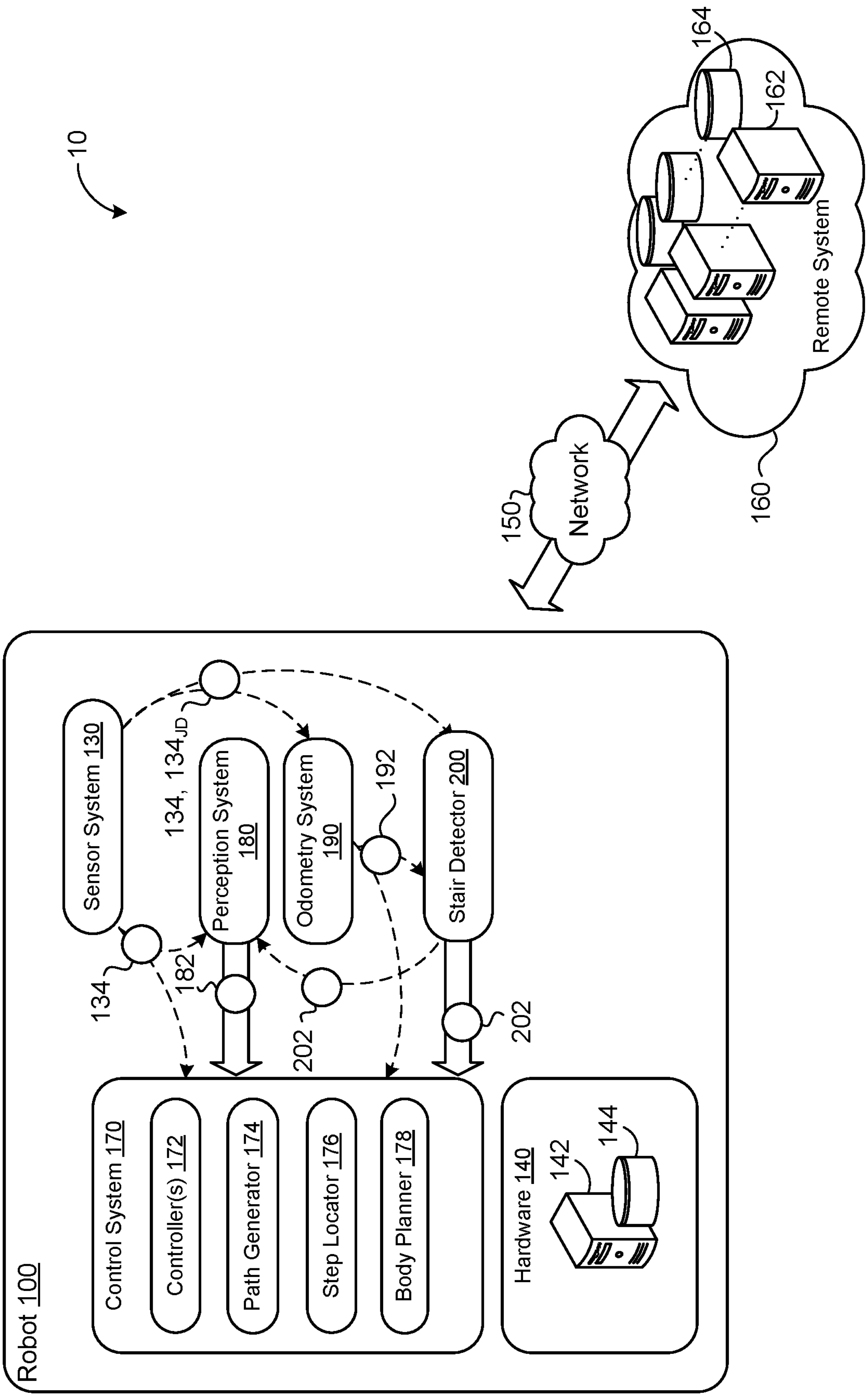
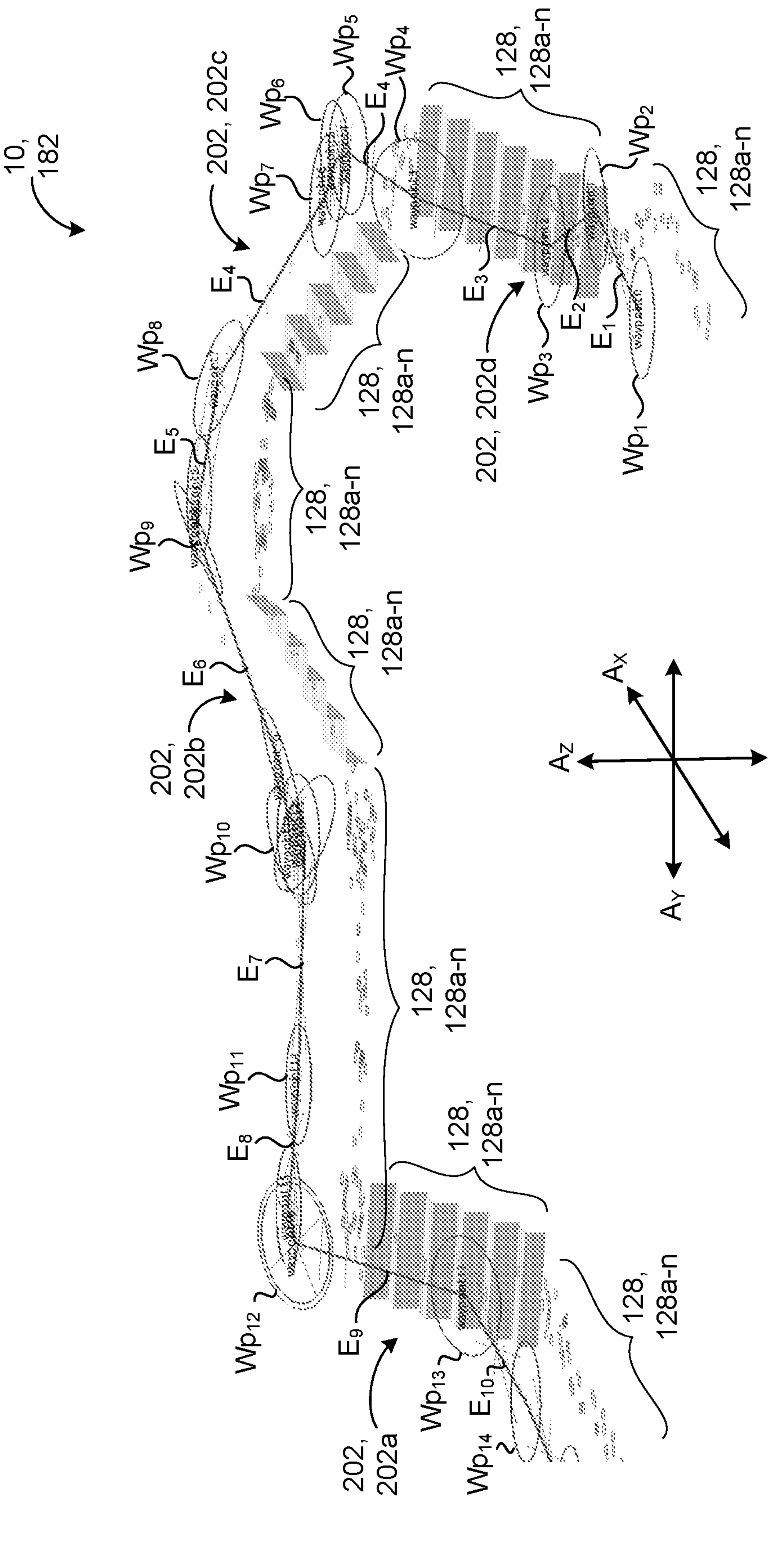

10
Robot 100
Sensor System 130
134
134, 134JD
Perception System 180
Odometry System 190
Stair Detector 200
182
192
202
202
Control System 170
Controller(s) 172
Path Generator 174
Step Locator 176
Body Planner 178
Hardware 140
142
144
Network
150
Remote System
160
162
164

FIG. 1B



10,
182
202, 202c
202, 202d
202,
202b
202,
202a
Wp1
Wp2
Wp3
Wp4
Wp5
Wp6
Wp7
Wp8
Wp9
Wp10
Wp11
Wp12
Wp13
Wp14
E1
E2
E3
E4
E5
E6
E7
E8
E9
E10
128,
128a-n
AX
AY
AZ

FIG. 1C

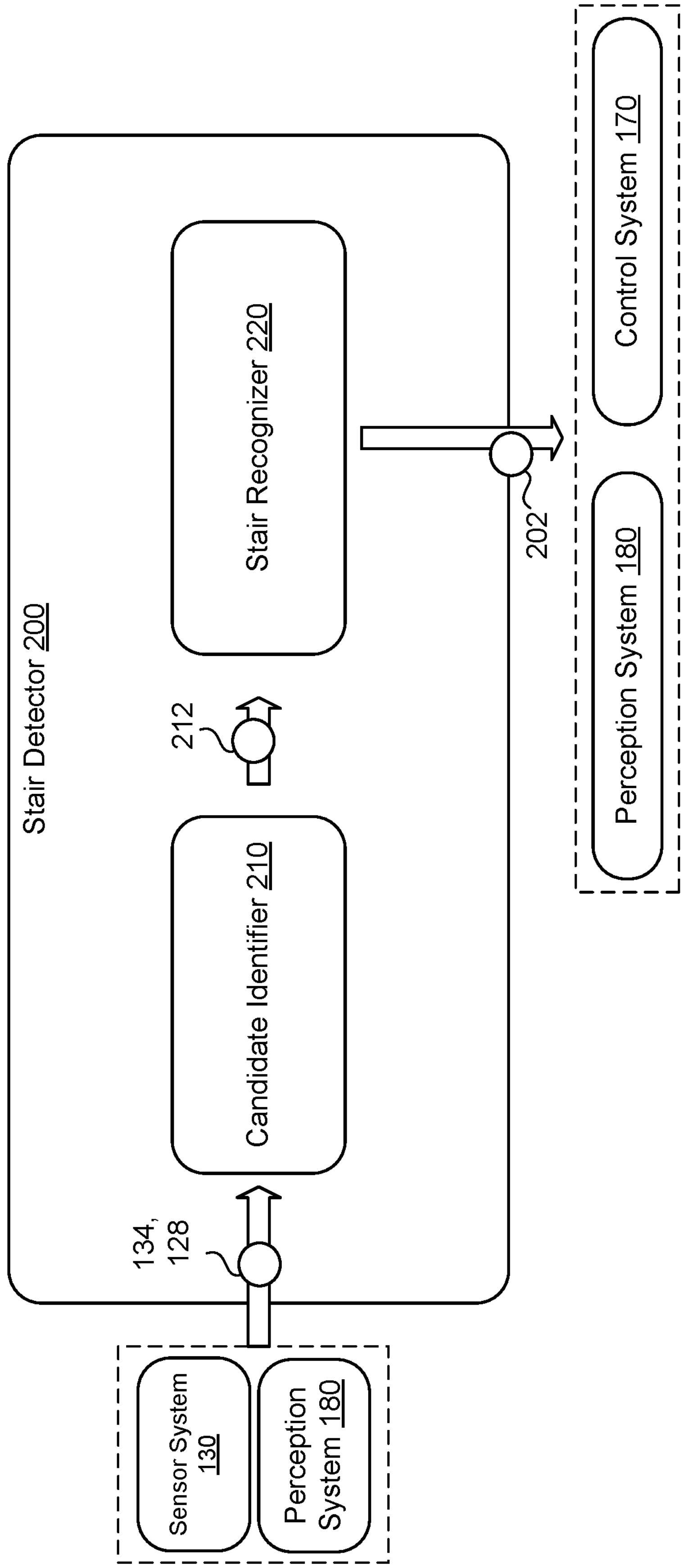
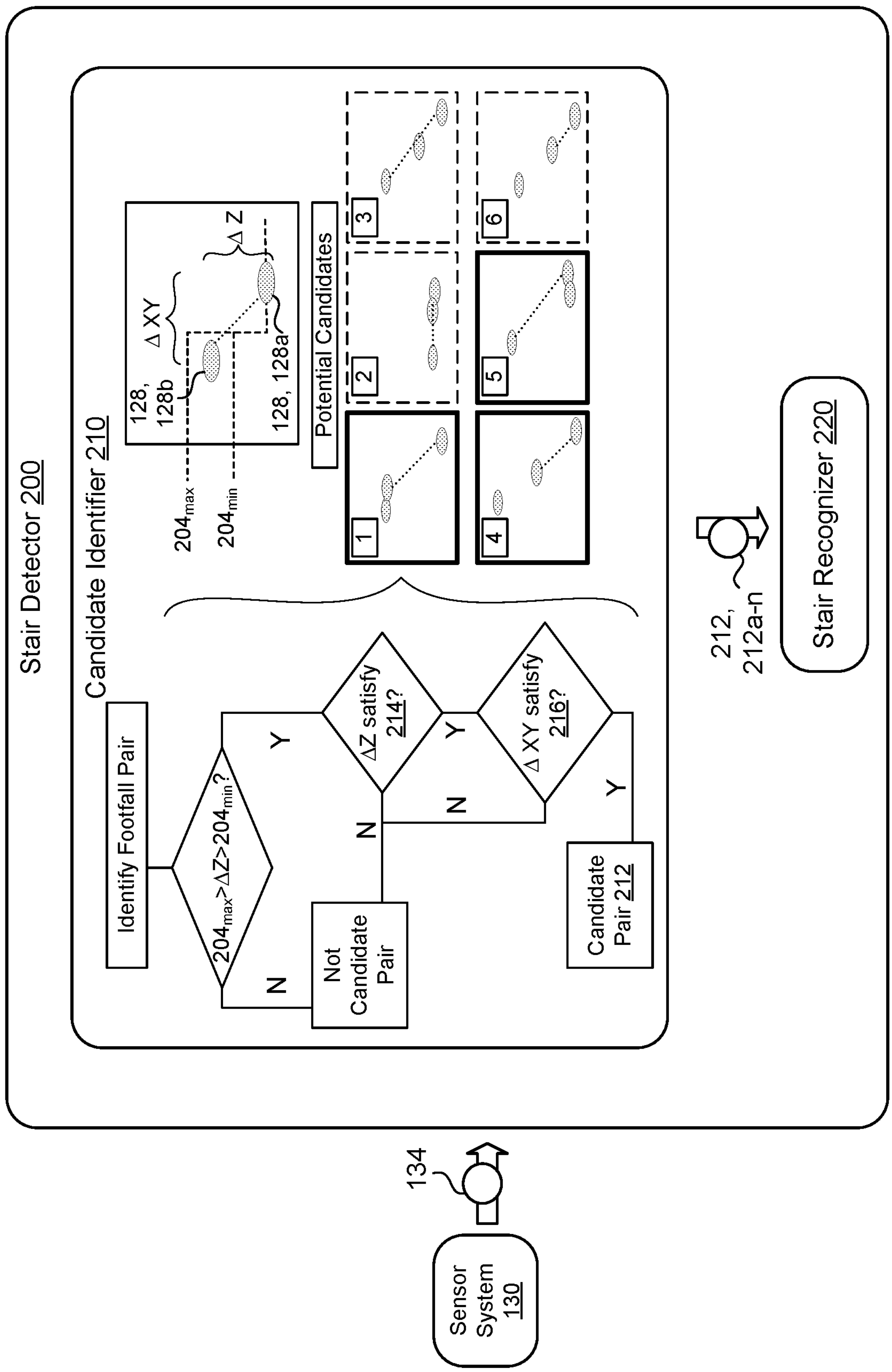
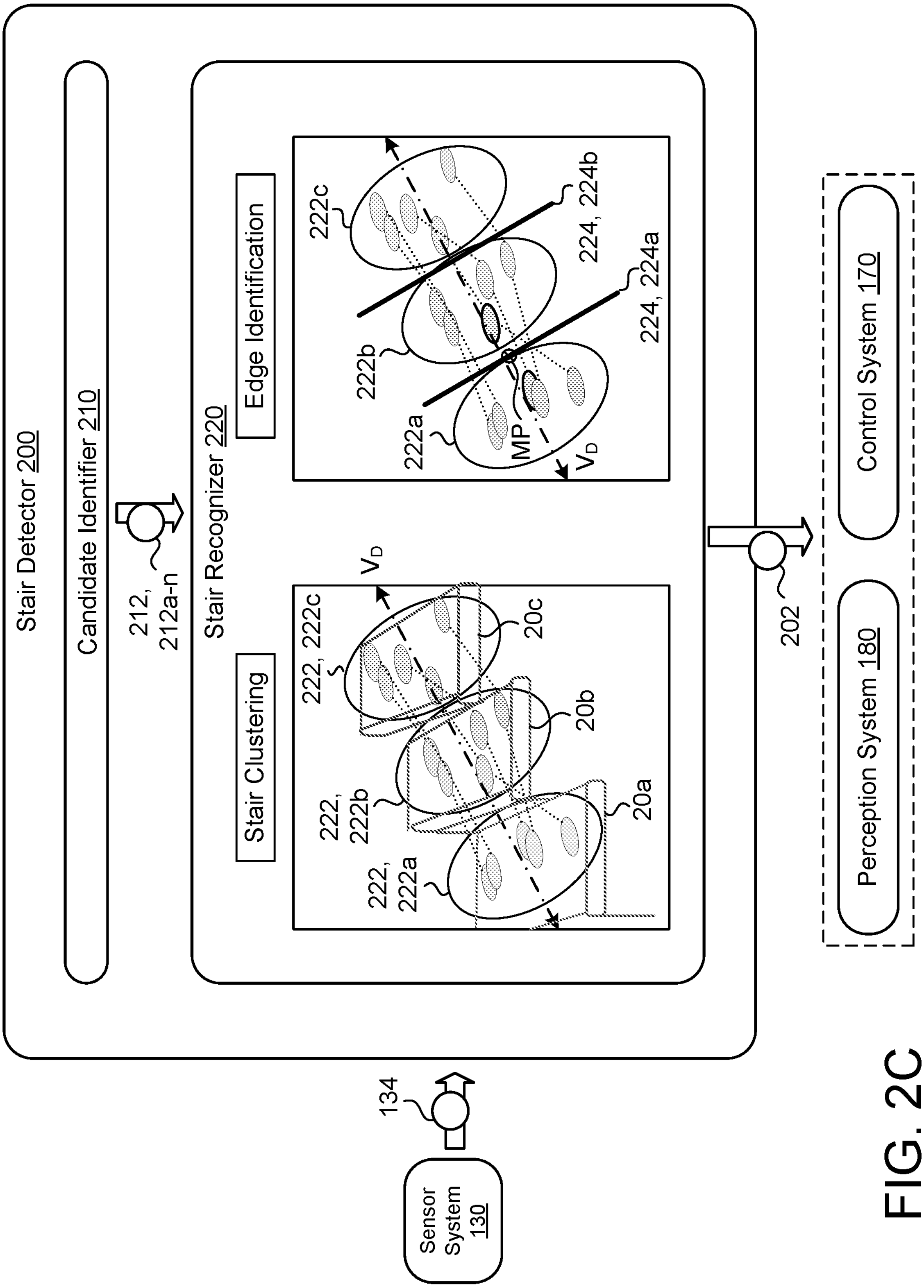

Stair Detector 200
Candidate Identifier 210
Stair Recognizer 220
212
134, 128
202
Sensor System 130
Perception System 180
Perception System 180
Control System 170

FIG. 2A



Stair Detector 200
Candidate Identifier 210
Identify Footfall Pair
$204_{max}>\Delta Z>204_{min}$?
N
Y
Not Candidate Pair
ΔZ satisfy 214?
N
Y
Δ XY satisfy 216?
Y
Candidate Pair 212
204_{max}
204_{min}
128, 128b
128, 128a
Δ XY
Δ Z
Potential Candidates
1
2
3
4
5
6
212, 212a-n
Stair Recognizer 220
134
Sensor System 130

FIG. 2B



Stair Detector 200
Candidate Identifier 210
212, 212a-n
Stair Recognizer 220
Stair Clustering
Edge Identification
222, 222a
222, 222b
222, 222c
20a
20b
20c
V_D
222a
222b
222c
MP
224, 224a
224, 224b
202
Perception System 180
Control System 170
134
Sensor System 130

FIG. 2C

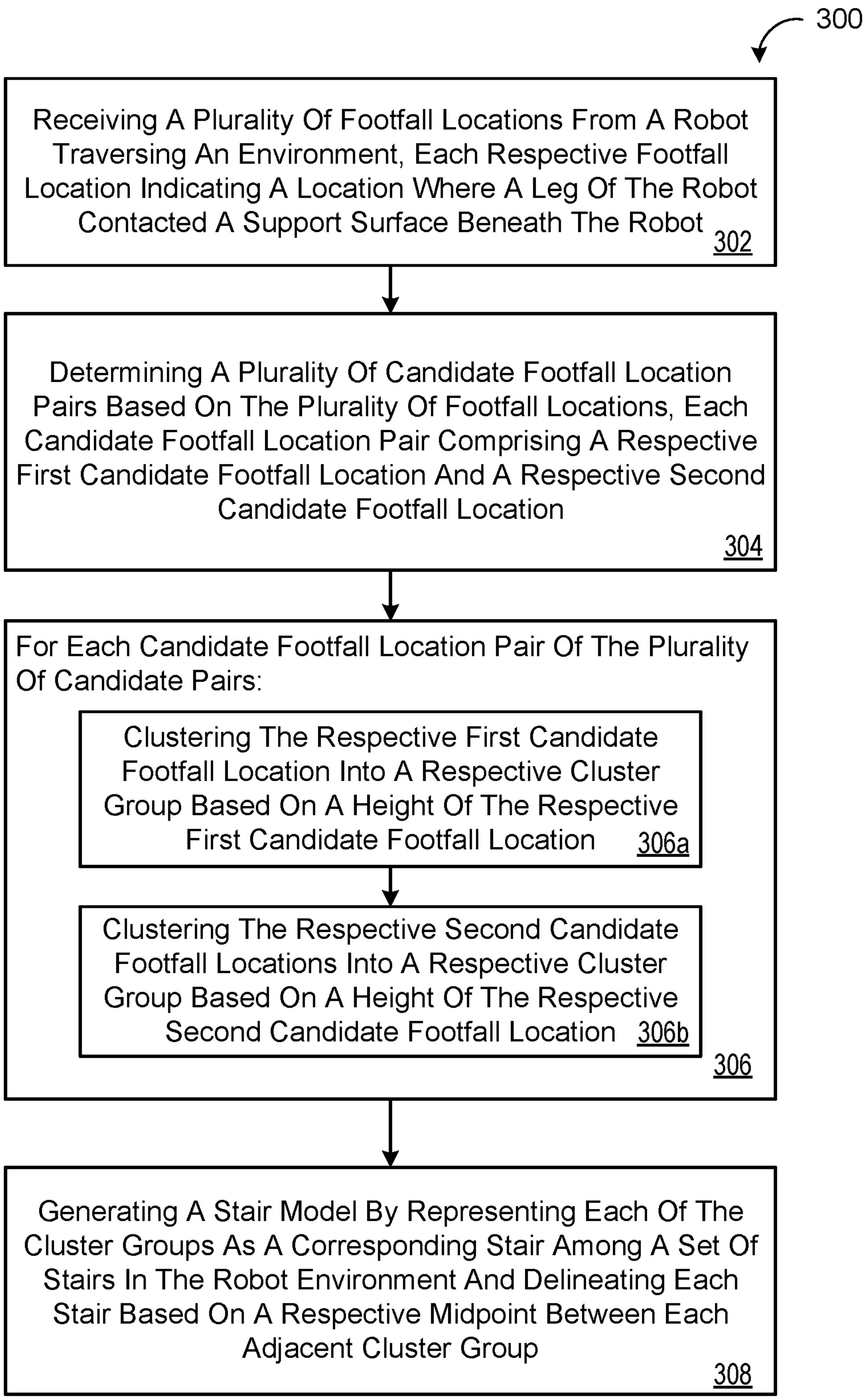

300
Receiving A Plurality Of Footfall Locations From A Robot Traversing An Environment, Each Respective Footfall Location Indicating A Location Where A Leg Of The Robot Contacted A Support Surface Beneath The Robot 302
Determining A Plurality Of Candidate Footfall Location Pairs Based On The Plurality Of Footfall Locations, Each Candidate Footfall Location Pair Comprising A Respective First Candidate Footfall Location And A Respective Second Candidate Footfall Location 304
For Each Candidate Footfall Location Pair Of The Plurality Of Candidate Pairs:
Clustering The Respective First Candidate Footfall Location Into A Respective Cluster Group Based On A Height Of The Respective First Candidate Footfall Location 306a
Clustering The Respective Second Candidate Footfall Locations Into A Respective Cluster Group Based On A Height Of The Respective Second Candidate Footfall Location 306b
306
Generating A Stair Model By Representing Each Of The Cluster Groups As A Corresponding Stair Among A Set Of Stairs In The Robot Environment And Delineating Each Stair Based On A Respective Midpoint Between Each Adjacent Cluster Group 308

FIG. 3

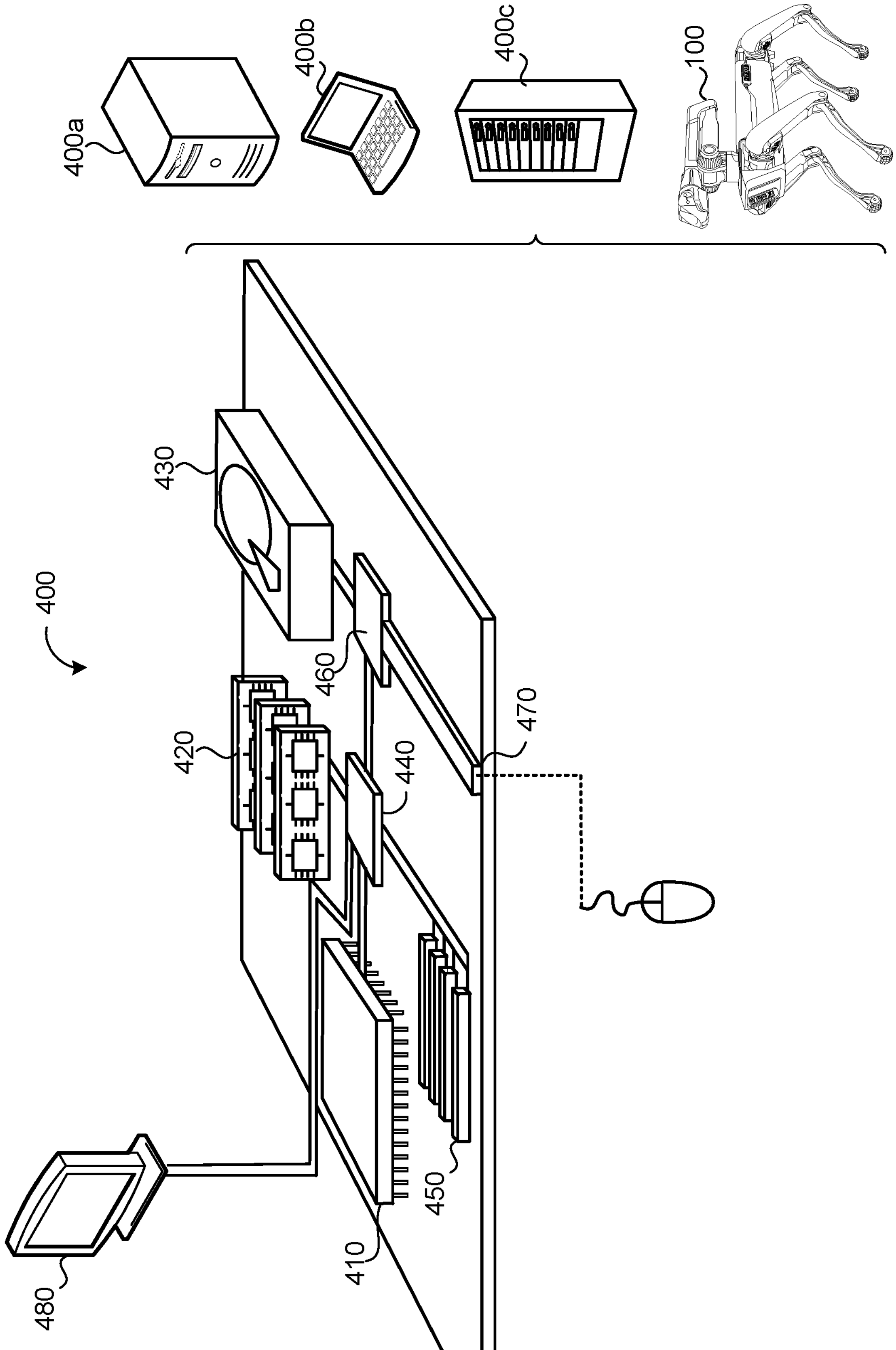


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2021/022953

A. CLASSIFICATION OF SUBJECT MATTER
INV. B62D57/024 B62D57/032
ADD.

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)
B62D

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2018/035320 A1 (UNIV PENNSYLVANIA [US]) 22 February 2018 (2018-02-22) figures 14-17 page 15, line 23 - page 19 -----	1-20
A	WO 2005/087452 A1 (SONY CORP [JP]; GUTMANN STEFFEN [JP]; FUKUCHI MASAKI [JP]) 22 September 2005 (2005-09-22) figures 1,8A-13B -----	1-20
A	US 9 989 970 B1 (MOREY CHRISTOPHER [US] ET AL) 5 June 2018 (2018-06-05) the whole document -----	1-20

☐ 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

"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
15 July 2021	23/07/2021

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Borges, Pedro

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2021/022953

Patent document cited in search report		Publication date	Patent family member(s)			Publication date
WO 2018035320	A1	22-02-2018	NONE			
WO 2005087452	A1	22-09-2005	JP	4618247	B2	26-01-2011
			JP	WO2005087452	A1	24-01-2008
			WO	2005087452	A1	22-09-2005
US 9989970	B1	05-06-2018	NONE			