US 20210373173A1

(19) United States
(12) Patent Application Publication
(10) Pub. No.: US 2021/0373173 A1
(43) Pub. Date: Dec. 2, 2021
Ozaslan

(54) **IDENTIFYING BACKGROUND FEATURES USING LIDAR**

(71) Applicant: **Motional AD LLC**, Boston, MA (US)

(72) Inventor: **Tolga Ozaslan**, Malden, MA (US)

(21) Appl. No.: **17/331,022**

(22) Filed: **May 26, 2021**

Related U.S. Application Data

(60) Provisional application No. 63/033,816, filed on Jun. 2, 2020.

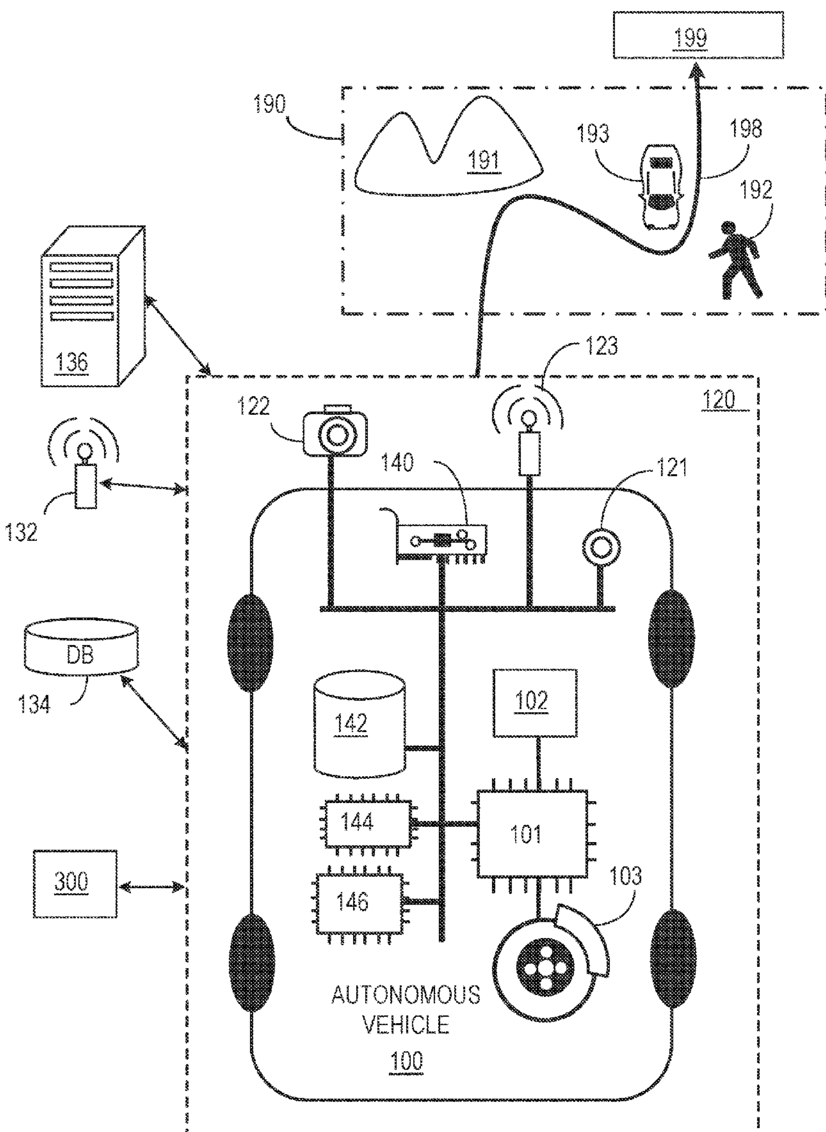
Publication Classification

(51) **Int. Cl.**
G01S 17/931 (2006.01)
G01S 17/89 (2006.01)

(52) **U.S. Cl.**
CPC ***G01S 17/931*** (2020.01); ***G01S 17/89*** (2013.01)

(57) **ABSTRACT**

Among other things, techniques are described for identifying background features using LiDAR. The techniques include modeling the point cloud information as a sphere and identifying faces corresponding to respective clusters of points of the received LiDAR point cloud information. A graph data structure is generated that includes vertices corresponding to respective faces of the identified faces and vertices of the graph data structure are connected based on adjacency of the respective underlying points and characteristics of the corresponding faces. Characteristics of the faces are analyzed corresponding to the vertices of the subgraph, and subgraphs are identified that correspond to a background feature of the physical environment based on the analysis.



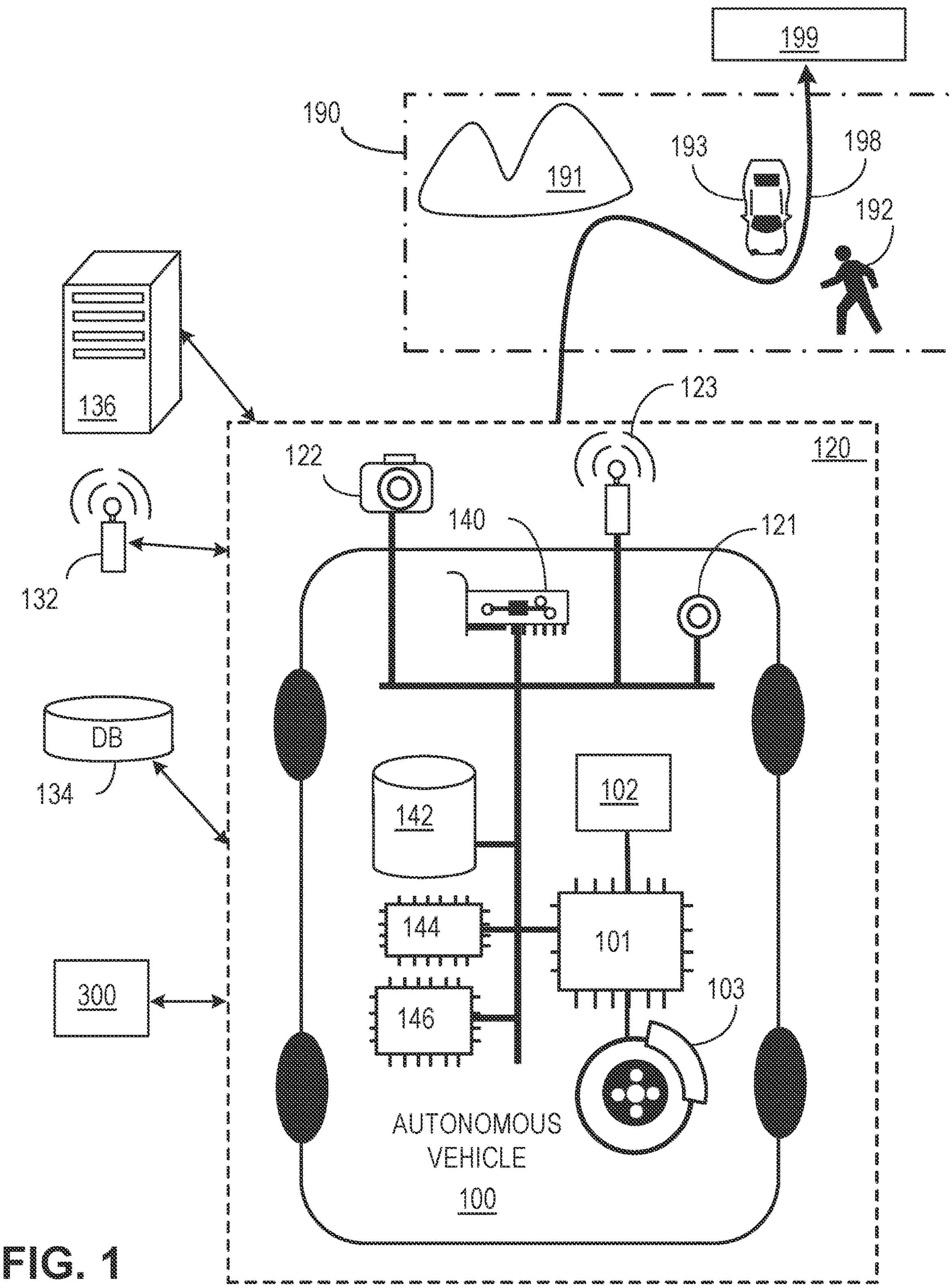

199
190
191
193
198
192
136
123
120
122
140
121
132
DB
134
142
102
144
101
146
103
300
AUTONOMOUS
VEHICLE
100

FIG. 1

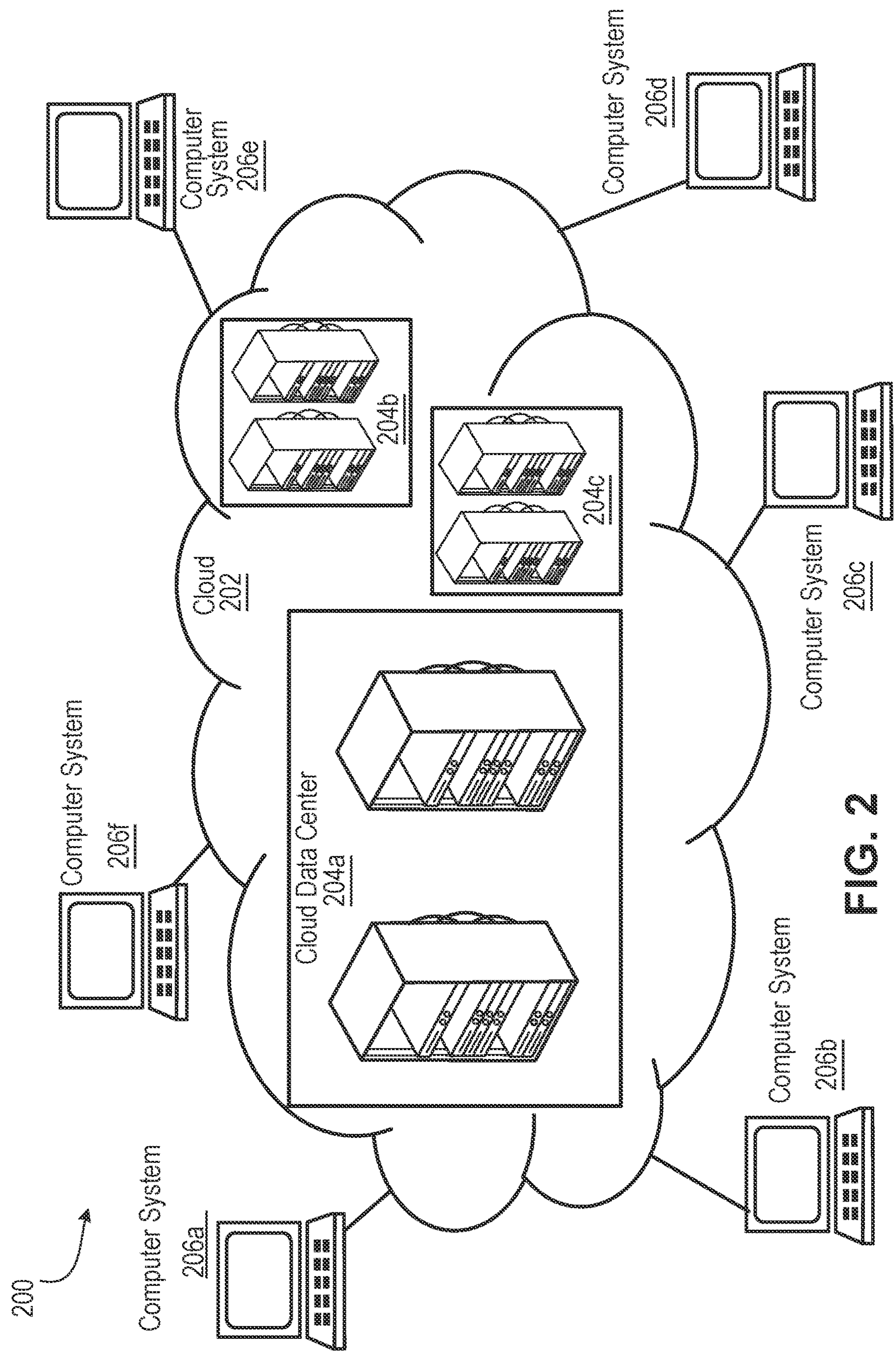

200
Computer System
206a
Computer System
206b
Computer System
206c
Computer System
206d
Computer System
206e
Computer System
206f
Cloud
202
Cloud Data Center
204a
204b
204c

FIG. 2

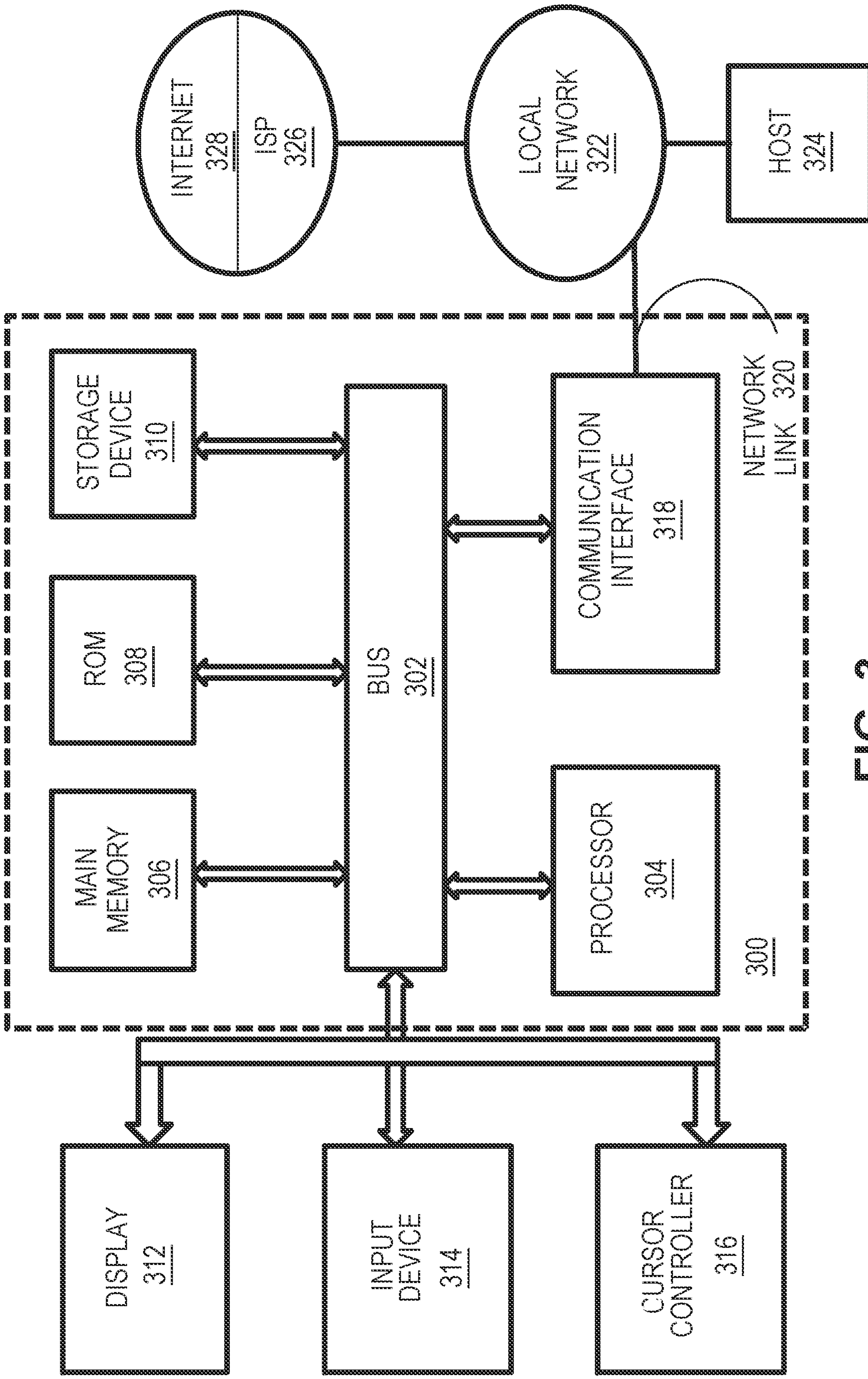

INTERNET
328
ISP
326
LOCAL
NETWORK
322
HOST
324
NETWORK
LINK 320
STORAGE
DEVICE
310
ROM
308
MAIN
MEMORY
306
BUS
302
COMMUNICATION
INTERFACE
318
PROCESSOR
304
300
DISPLAY
312
INPUT
DEVICE
314
CURSOR
CONTROLLER
316

FIG. 3

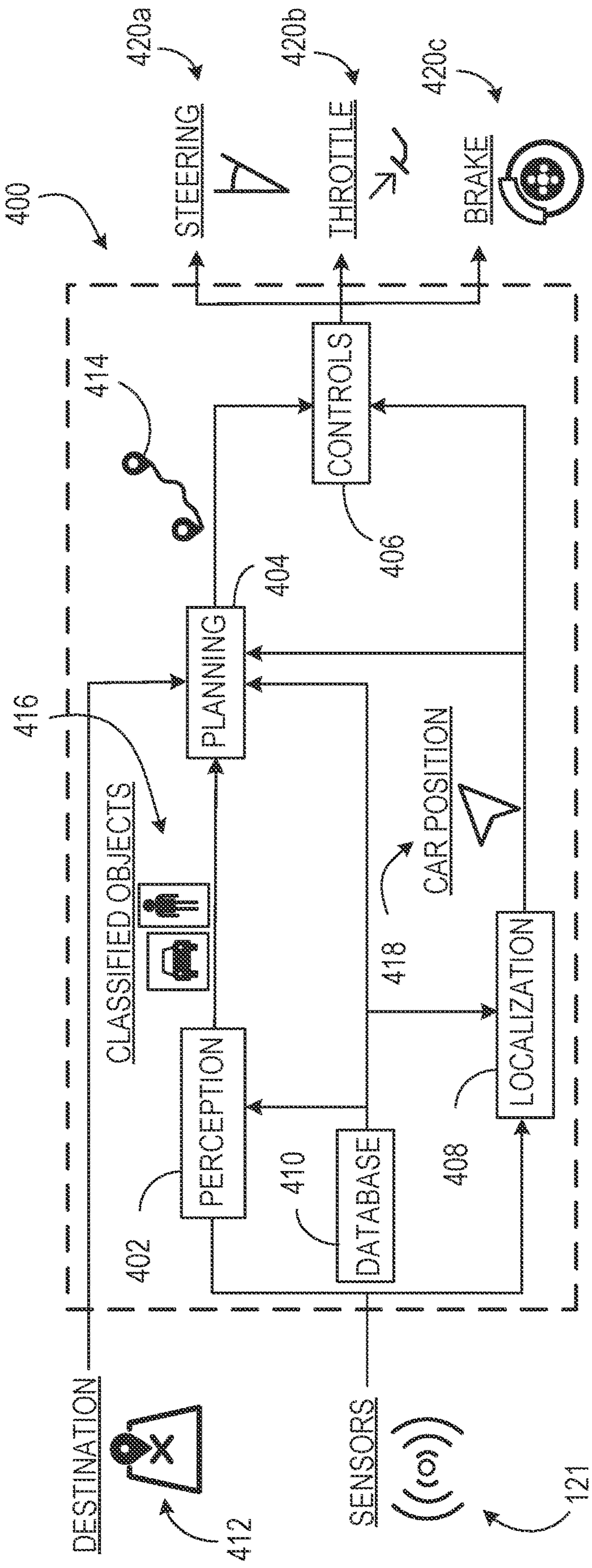

400
STEERING
420a
THROTTLE
420b
BRAKE
420c
CONTROLS
406
414
PLANNING
404
416
CLASSIFIED OBJECTS
CAR POSITION
418
PERCEPTION
402
DATABASE
410
LOCALIZATION
408
DESTINATION
412
SENSORS
121

FIG. 4

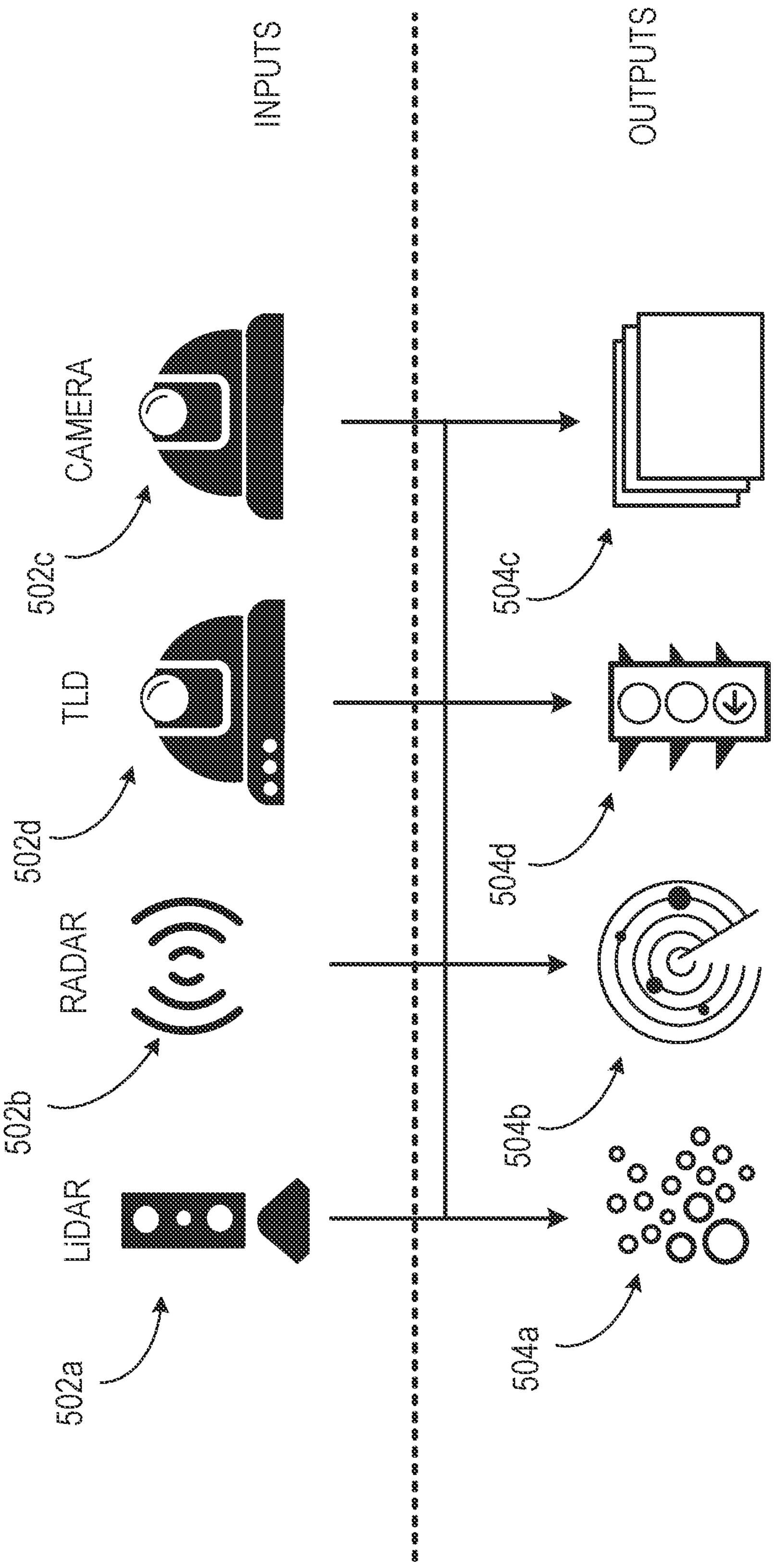

INPUTS
OUTPUTS
502a
LIDAR
502b
RADAR
502d
TLD
502c
CAMERA
504a
504b
504d
504c

FIG. 5

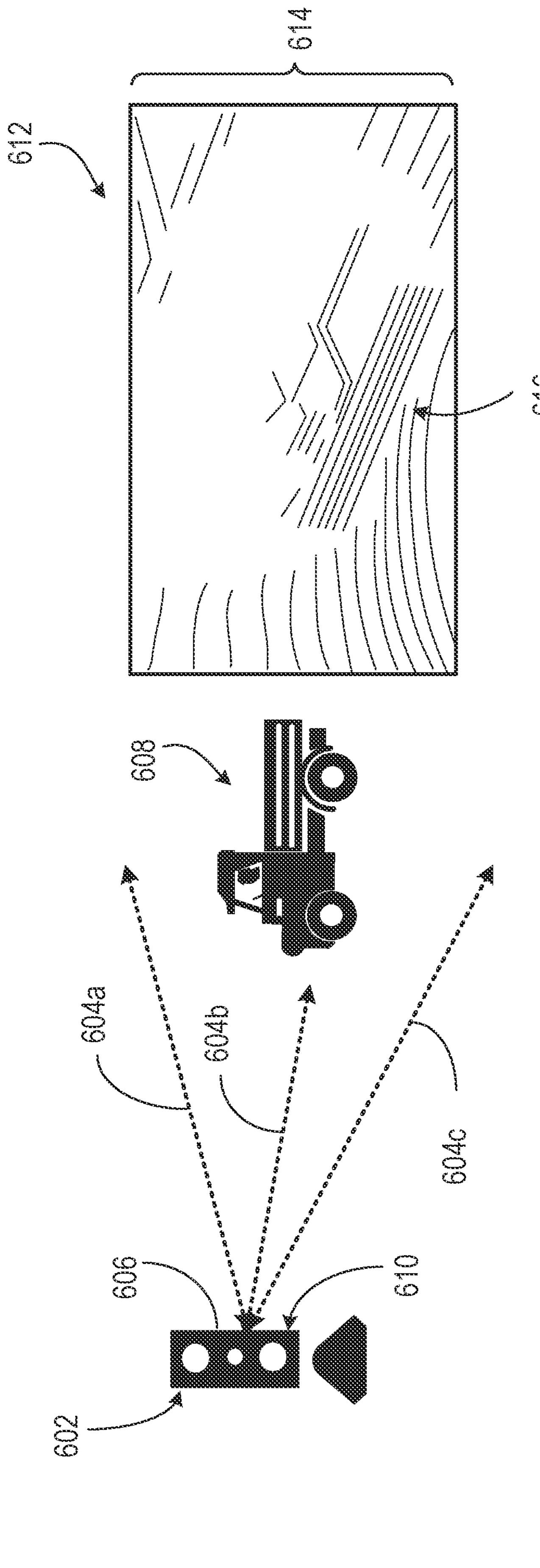


FIG. 6

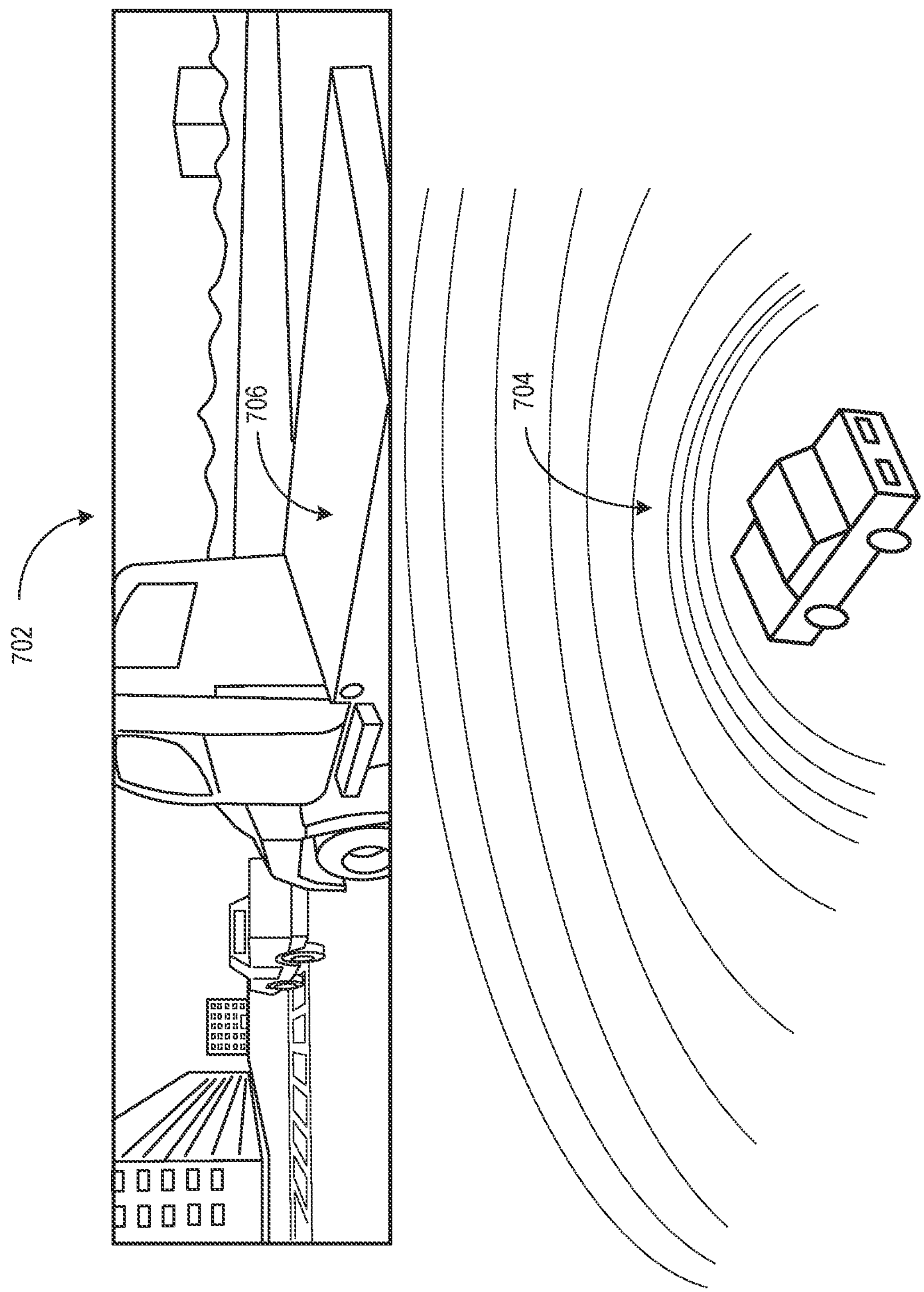


FIG. 7

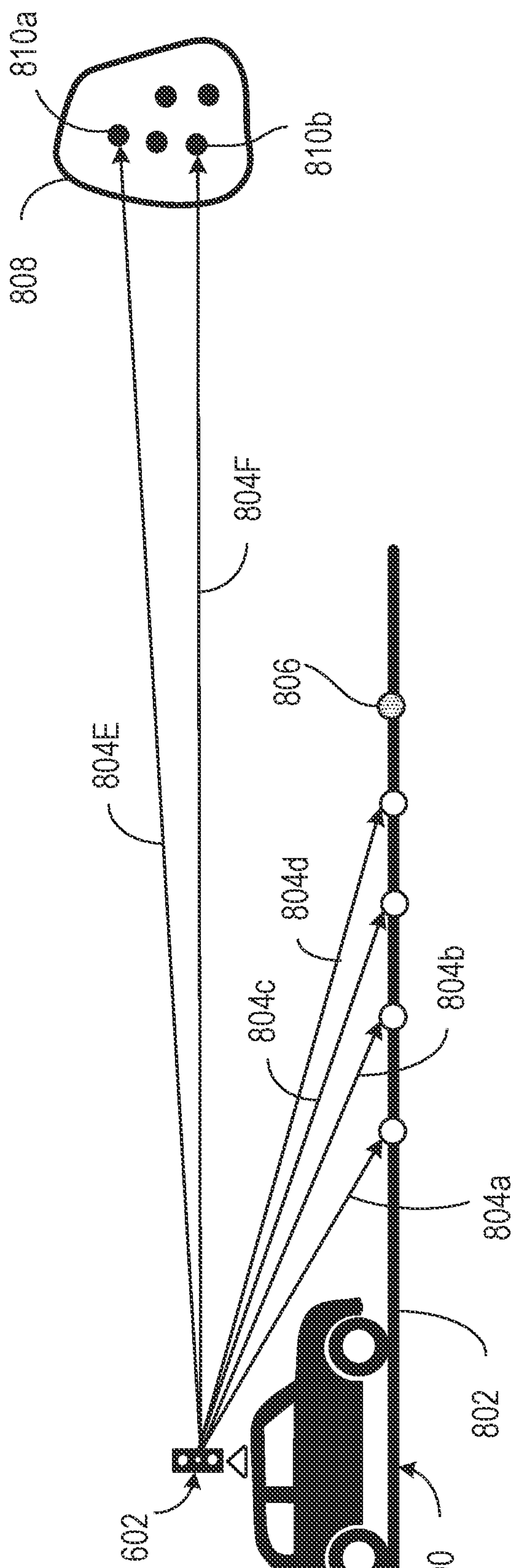


FIG. 8

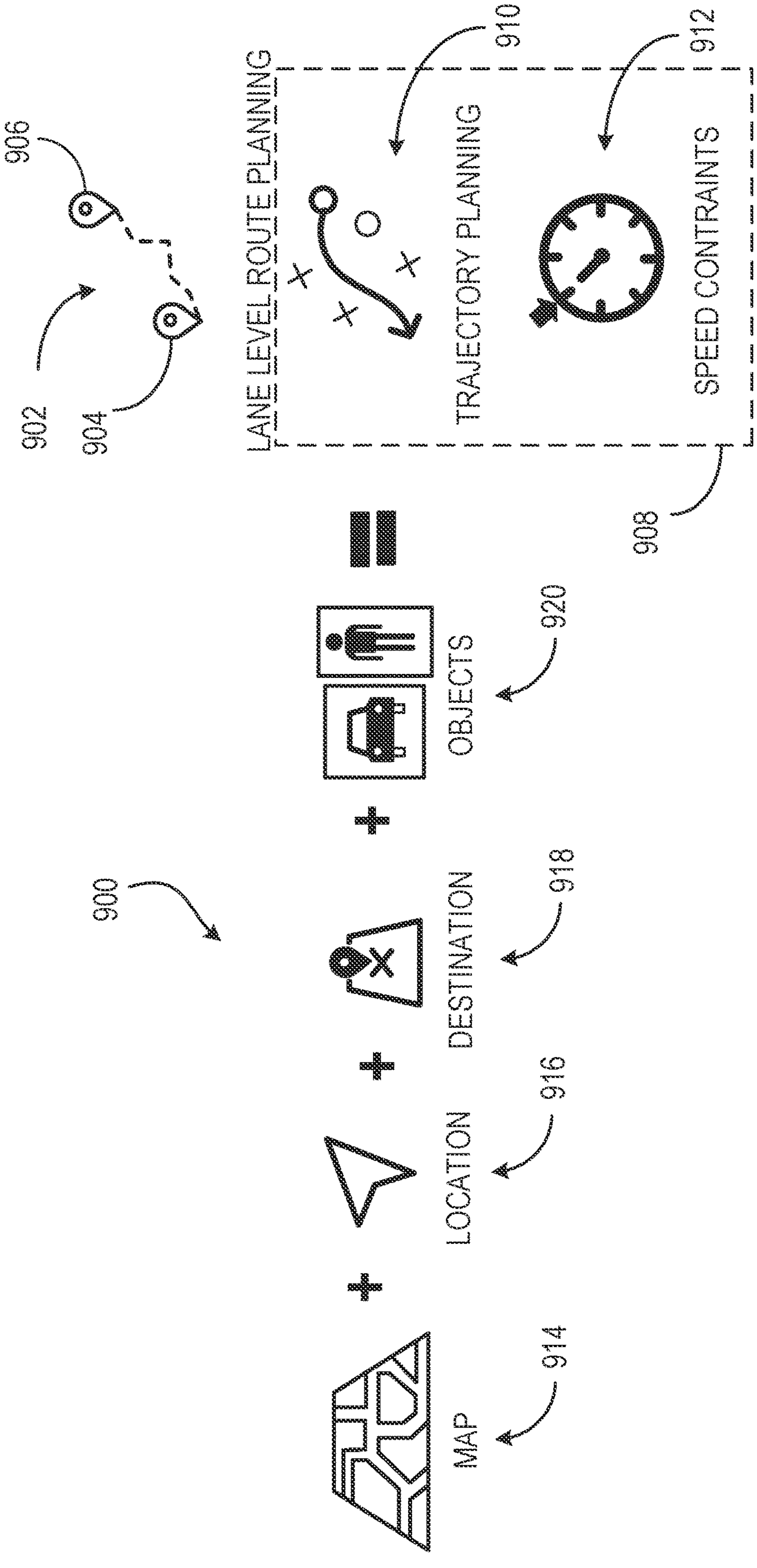

900
902
904
906
LANE LEVEL ROUTE PLANNING
910
TRAJECTORY PLANNING
912
SPEED CONTRAINTS
908
MAP
914
LOCATION
916
DESTINATION
918
OBJECTS
920

FIG. 9

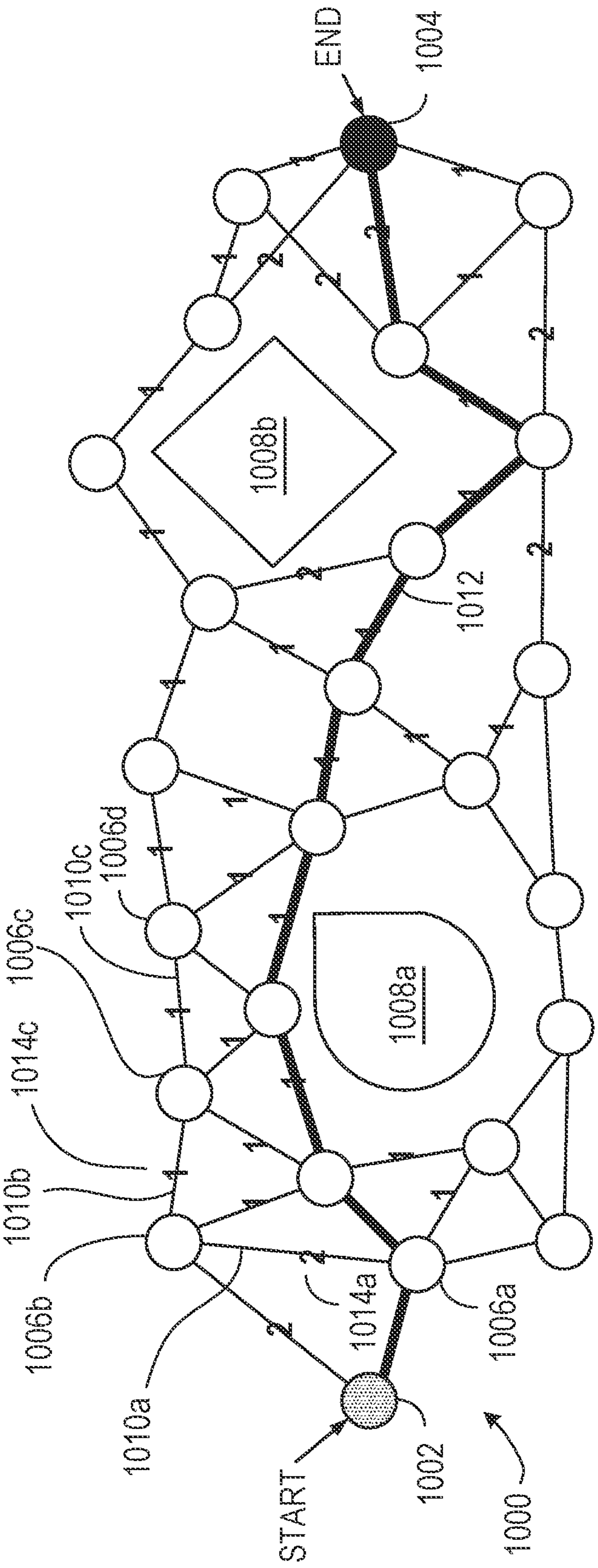

END
1004
1008b
1012
1010c
1006d
1006c
1014c
1010b
1008a
1006b
1014a
1006a
1010a
START
1002
1000

FIG. 10

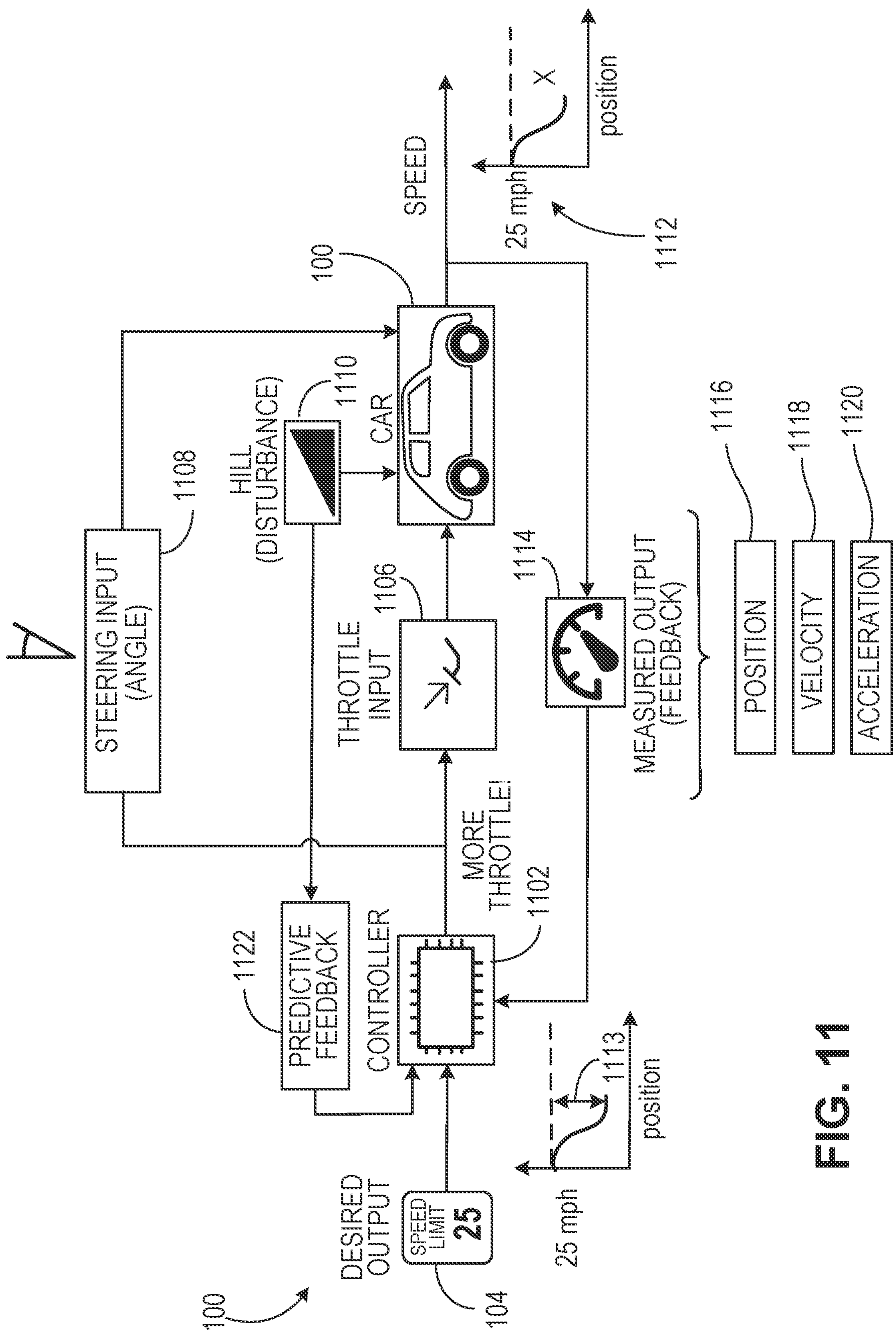

1100
DESIRED OUTPUT
SPEED LIMIT 25
1104
25 mph
position
1113
CONTROLLER
1102
PREDICTIVE FEEDBACK
1122
MORE THROTTLE!
STEERING INPUT (ANGLE)
1108
HILL (DISTURBANCE)
1110
THROTTLE INPUT
1106
CAR
100
MEASURED OUTPUT (FEEDBACK)
1114
POSITION
1116
VELOCITY
1118
ACCELERATION
1120
SPEED
25 mph
X
position
1112

FIG. 11

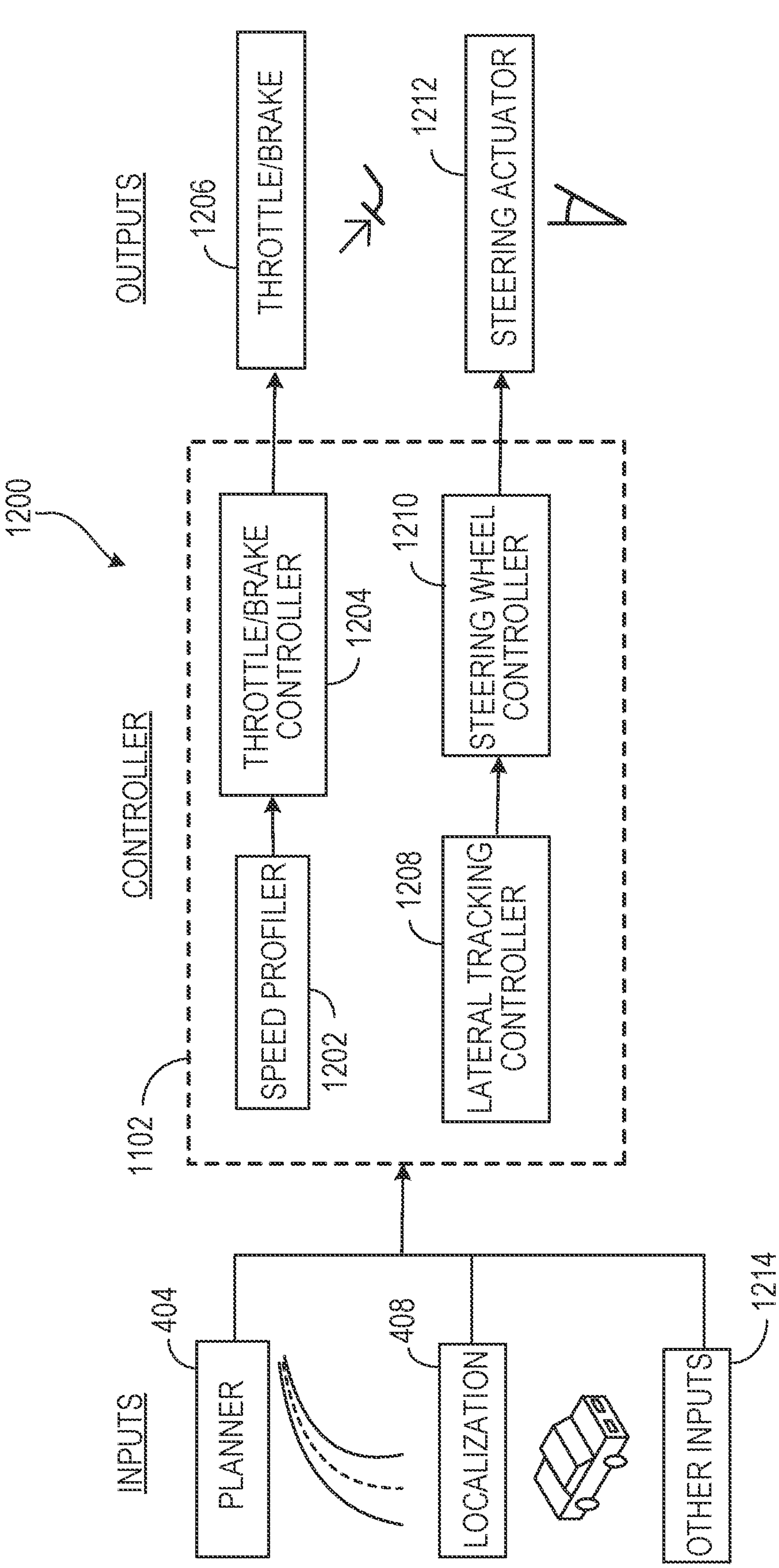
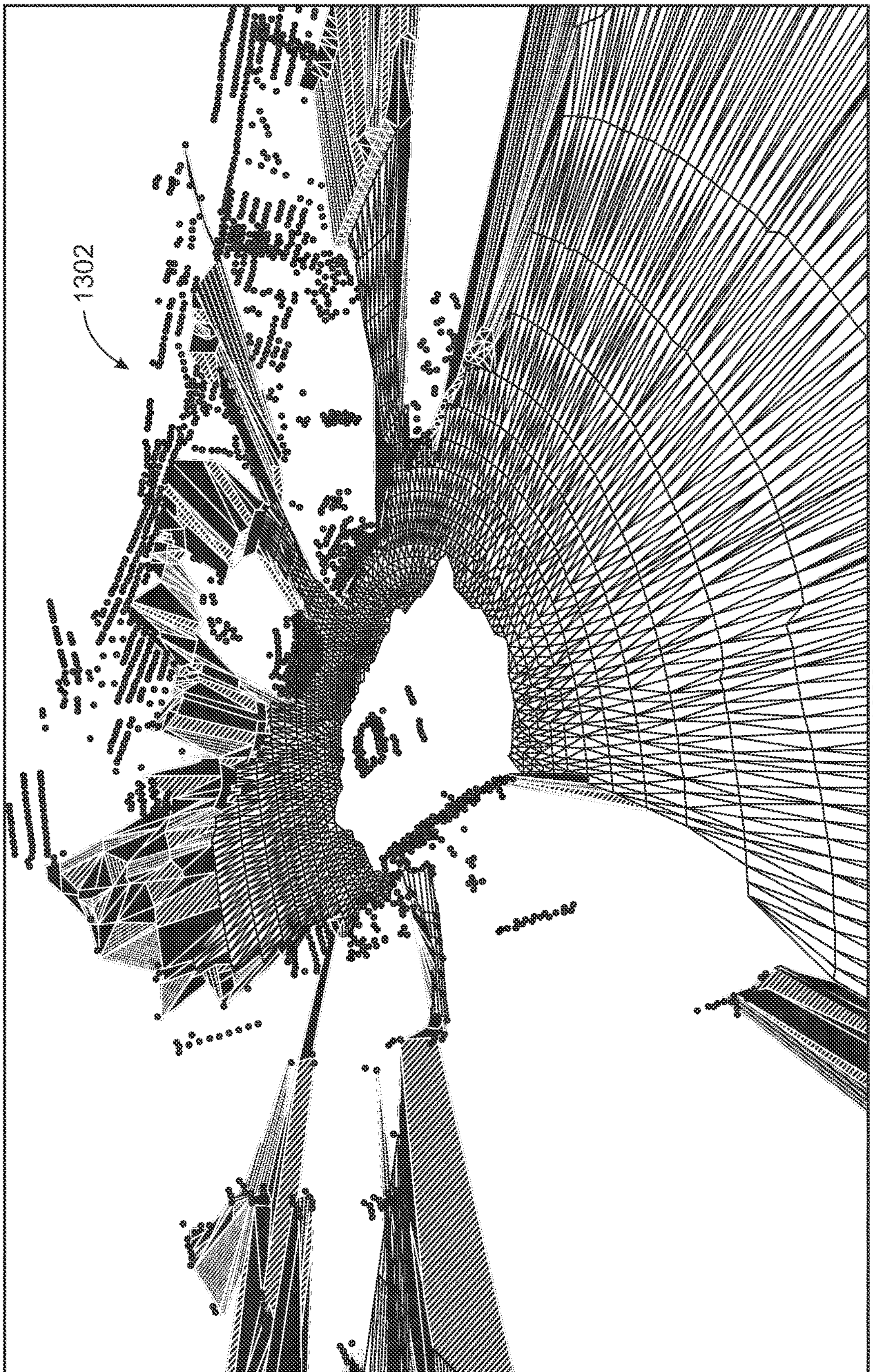
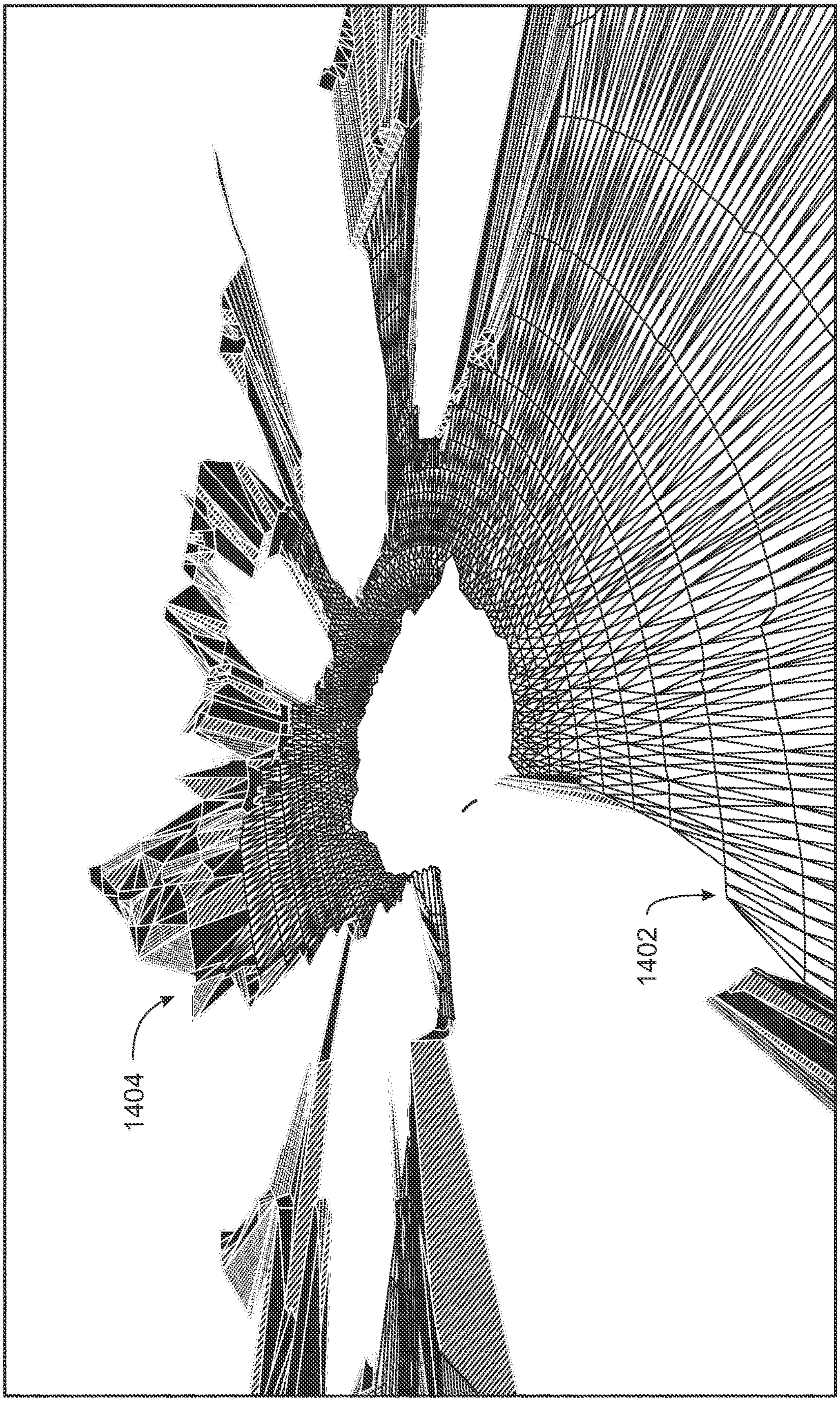

1200
INPUTS
PLANNER
404
LOCALIZATION
408
OTHER INPUTS
1214
CONTROLLER
1102
SPEED PROFILER
1202
THROTTLE/BRAKE CONTROLLER
1204
LATERAL TRACKING CONTROLLER
1208
STEERING WHEEL CONTROLLER
1210
OUTPUTS
THROTTLE/BRAKE
1206
STEERING ACTUATOR
1212

FIG. 12



1300
FIG. 13



1400
FIG. 14

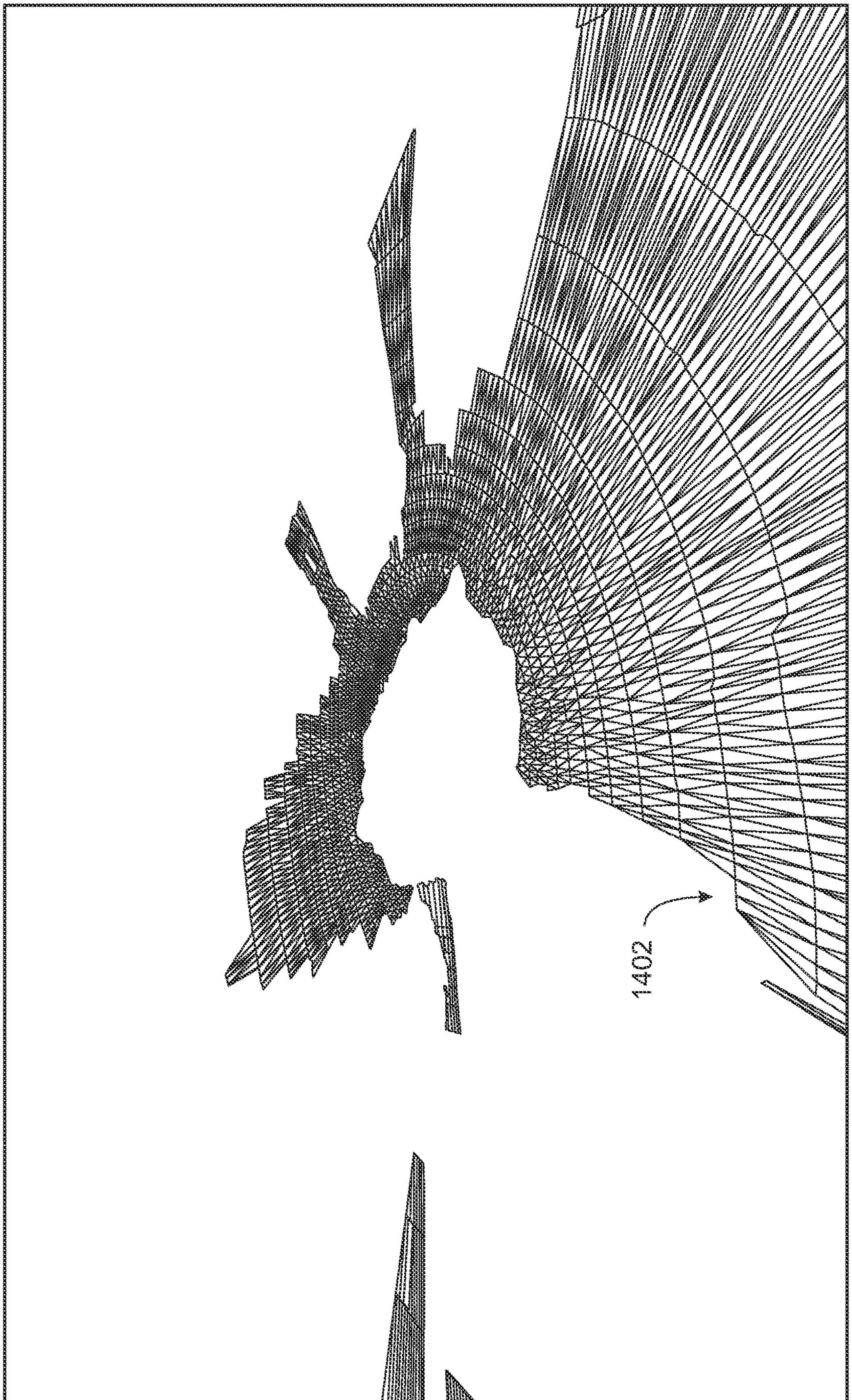
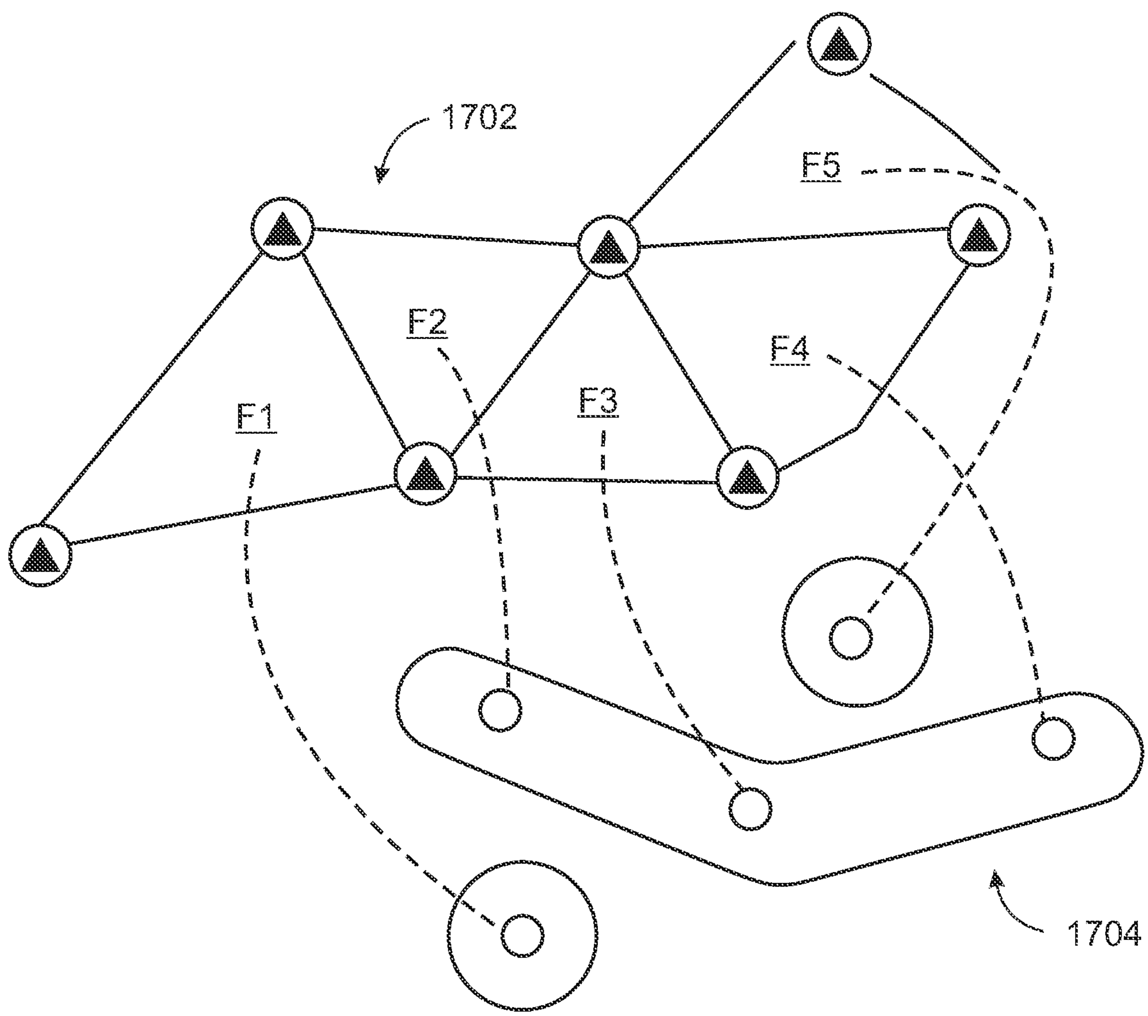


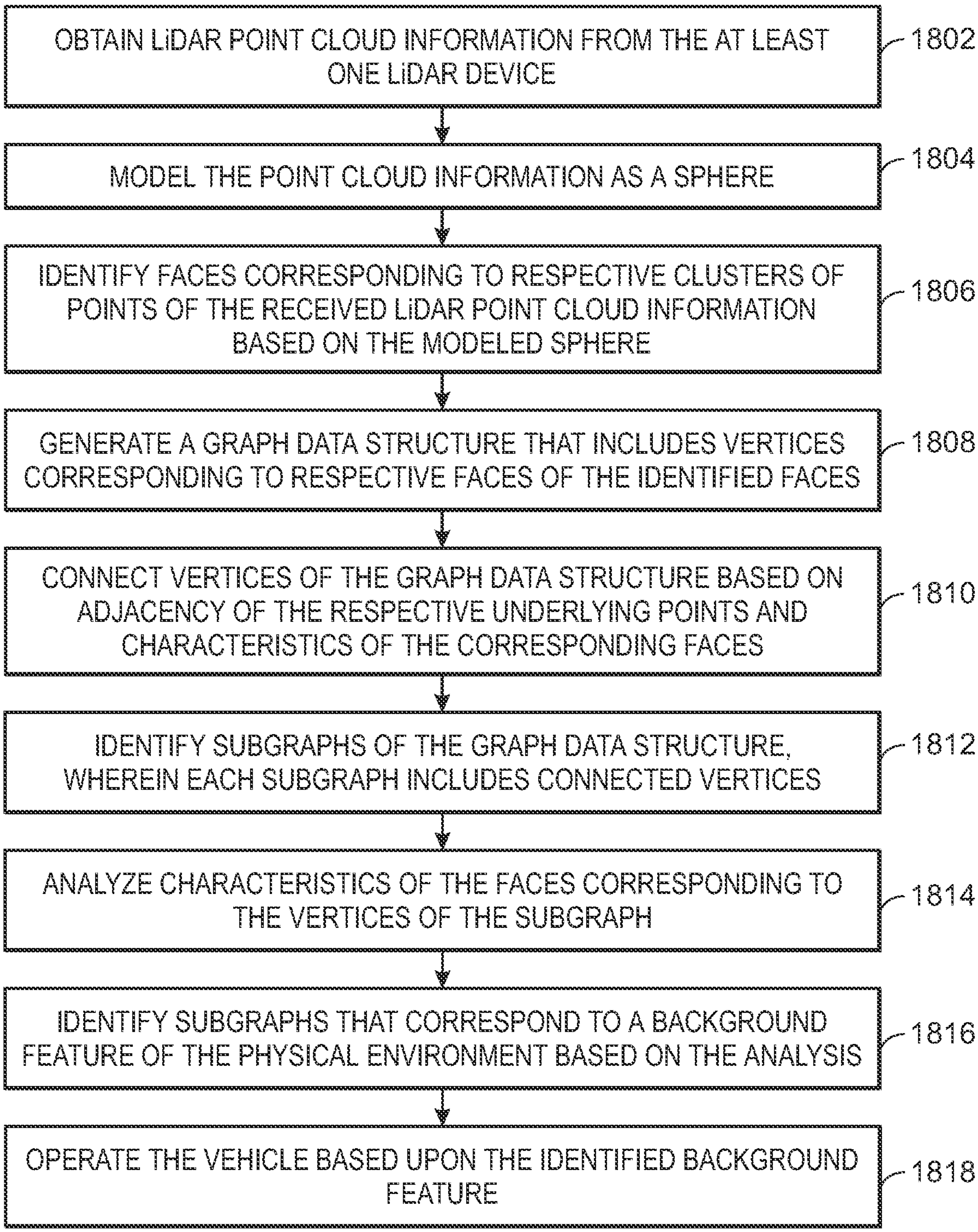
FIG. 15

FIG. 16



1700

FIG. 17


OBTAIN LiDAR POINT CLOUD INFORMATION FROM THE AT LEAST ONE LiDAR DEVICE
1802
MODEL THE POINT CLOUD INFORMATION AS A SPHERE
1804
IDENTIFY FACES CORRESPONDING TO RESPECTIVE CLUSTERS OF POINTS OF THE RECEIVED LiDAR POINT CLOUD INFORMATION BASED ON THE MODELED SPHERE
1806
GENERATE A GRAPH DATA STRUCTURE THAT INCLUDES VERTICES CORRESPONDING TO RESPECTIVE FACES OF THE IDENTIFIED FACES
1808
CONNECT VERTICES OF THE GRAPH DATA STRUCTURE BASED ON ADJACENCY OF THE RESPECTIVE UNDERLYING POINTS AND CHARACTERISTICS OF THE CORRESPONDING FACES
1810
IDENTIFY SUBGRAPHS OF THE GRAPH DATA STRUCTURE, WHEREIN EACH SUBGRAPH INCLUDES CONNECTED VERTICES
1812
ANALYZE CHARACTERISTICS OF THE FACES CORRESPONDING TO THE VERTICES OF THE SUBGRAPH
1814
IDENTIFY SUBGRAPHS THAT CORRESPOND TO A BACKGROUND FEATURE OF THE PHYSICAL ENVIRONMENT BASED ON THE ANALYSIS
1816
OPERATE THE VEHICLE BASED UPON THE IDENTIFIED BACKGROUND FEATURE
1818

1800

FIG. 18

IDENTIFYING BACKGROUND FEATURES USING LIDAR

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority to U.S. Provisional Patent Application No. 63/033,816, filed Jun. 2, 2020, the entire contents of which is incorporated herein by reference.

FIELD OF THE INVENTION

[0002] This description relates to identifying background features using LiDAR.

BACKGROUND

[0003] Vehicles (such as autonomous vehicles) can use LiDAR to identify features of the surrounding environment. LiDAR uses one or more emitters to illuminate nearby objects.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. **1** shows an example of an autonomous vehicle having autonomous capability.

[0005] FIG. **2** illustrates an example "cloud" computing environment.

[0006] FIG. **3** illustrates a computer system.

[0007] FIG. **4** shows an example architecture for an autonomous vehicle.

[0008] FIG. **5** shows an example of inputs and outputs that may be used by a perception module.

[0009] FIG. **6** shows an example of a LiDAR system.

[0010] FIG. **7** shows the LiDAR system in operation.

[0011] FIG. **8** shows the operation of the LiDAR system in additional detail.

[0012] FIG. **9** shows a block diagram of the relationships between inputs and outputs of a planning module.

[0013] FIG. **10** shows a directed graph used in path planning.

[0014] FIG. **11** shows a block diagram of the inputs and outputs of a control module.

[0015] FIG. **12** shows a block diagram of the inputs, outputs, and components of a controller.

[0016] FIG. **13** shows a partial mesh from a raw point cloud.

[0017] FIG. **14** shows faces representing the ground surface as well as other objects.

[0018] FIG. **15** shows only the identified ground surface.

[0019] FIG. **16** shows only the non-ground faces.

[0020] FIG. **17** shows the correspondence of faces f**1**-f**5** and a graph data structure.

[0021] FIG. **18** is a process flow diagram of a process for identifying background features using LiDAR.

DETAILED DESCRIPTION

[0022] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0023] In the drawings, specific arrangements or orderings of schematic elements, such as those representing devices, modules, instruction blocks and data elements, are shown for ease of description. However, it should be understood by those skilled in the art that the specific ordering or arrangement of the schematic elements in the drawings is not meant to imply that a particular order or sequence of processing, or separation of processes, is required. Further, the inclusion of a schematic element in a drawing is not meant to imply that such element is required in all embodiments or that the features represented by such element may not be included in or combined with other elements in some embodiments.

[0024] Further, in the drawings, where connecting elements, such as solid or dashed lines or arrows, are used to illustrate a connection, relationship, or association between or among two or more other schematic elements, the absence of any such connecting elements is not meant to imply that no connection, relationship, or association can exist. In other words, some connections, relationships, or associations between elements are not shown in the drawings so as not to obscure the disclosure. In addition, for ease of illustration, a single connecting element is used to represent multiple connections, relationships or associations between elements. For example, where a connecting element represents a communication of signals, data, or instructions, it should be understood by those skilled in the art that such element represents one or multiple signal paths (e.g., a bus), as may be needed, to affect the communication.

[0025] Reference will now be made in detail to embodiments, examples of which are illustrated in the accompanying drawings. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the various described embodiments. However, it will be apparent to one of ordinary skill in the art that the various described embodiments may be practiced without these specific details. In other instances, well-known methods, procedures, components, circuits, and networks have not been described in detail so as not to unnecessarily obscure aspects of the embodiments.

[0026] Several features are described hereafter that can each be used independently of one another or with any combination of other features. However, any individual feature may not address any of the problems discussed above or might only address one of the problems discussed above. Some of the problems discussed above might not be fully addressed by any of the features described herein. Although headings are provided, information related to a particular heading, but not found in the section having that heading, may also be found elsewhere in this description. Embodiments are described herein according to the following outline:

[0027] 1. General Overview

[0028] 2. System Overview

[0029] 3. Autonomous Vehicle Architecture

[0030] 4. Autonomous Vehicle Inputs

[0031] 5. Autonomous Vehicle Planning

[0032] 6. Autonomous Vehicle Control

[0033] 7. Background Detection

General Overview

[0034] A vehicle (such as an autonomous vehicle) can identify the ground in its environment by processing LiDAR point clouds to identify surfaces and determining which adjacent surfaces correspond to the ground. For example, the vehicle can generate a graph data structure, where each vertex of the graph corresponds to a portion of point cloud data, and identify adjacent vertices that share similar characteristics.

[0035] Some of the advantages include the identification of the ground in a manner that consumes less processing power and energy than other techniques. LiDAR data is modeled as a sphere rather than a cylinder which enables easier identification of adjacent surfaces. Data from multiple LiDAR devices can be used and does not necessarily need to be merged in advance. The point cloud data does not need to be provided in a particular order.

System Overview

[0036] FIG. **1** shows an example of an autonomous vehicle **100** having autonomous capability.

[0037] As used herein, the term "autonomous capability" refers to a function, feature, or facility that enables a vehicle to be partially or fully operated without real-time human intervention, including without limitation fully autonomous vehicles, highly autonomous vehicles, and conditionally autonomous vehicles.

[0038] As used herein, an autonomous vehicle (AV) is a vehicle that possesses autonomous capability.

[0039] As used herein, "vehicle" includes means of transportation of goods or people. For example, cars, buses, trains, airplanes, drones, trucks, boats, ships, submersibles, dirigibles, etc. A driverless car is an example of a vehicle.

[0040] As used herein, "trajectory" refers to a path or route to navigate an AV from a first spatiotemporal location to second spatiotemporal location. In an embodiment, the first spatiotemporal location is referred to as the initial or starting location and the second spatiotemporal location is referred to as the destination, final location, goal, goal position, or goal location. In some examples, a trajectory is made up of one or more segments (e.g., sections of road) and each segment is made up of one or more blocks (e.g., portions of a lane or intersection). In an embodiment, the spatiotemporal locations correspond to real world locations. For example, the spatiotemporal locations are pick up or drop-off locations to pick up or drop-off persons or goods.

[0041] As used herein, "sensor(s)" includes one or more hardware components that detect information about the environment surrounding the sensor. Some of the hardware components can include sensing components (e.g., image sensors, biometric sensors), transmitting and/or receiving components (e.g., laser or radio frequency wave transmitters and receivers), electronic components such as analog-to-digital converters, a data storage device (such as a RAM and/or a nonvolatile storage), software or firmware components and data processing components such as an ASIC (application-specific integrated circuit), a microprocessor and/or a microcontroller.

[0042] As used herein, a "scene description" is a data structure (e.g., list) or data stream that includes one or more classified or labeled objects detected by one or more sensors on the AV vehicle or provided by a source external to the AV.

[0043] As used herein, a "road" is a physical area that can be traversed by a vehicle, and may correspond to a named thoroughfare (e.g., city street, interstate freeway, etc.) or may correspond to an unnamed thoroughfare (e.g., a driveway in a house or office building, a section of a parking lot, a section of a vacant lot, a dirt path in a rural area, etc.). Because some vehicles (e.g., 4-wheel-drive pickup trucks, sport utility vehicles, etc.) are capable of traversing a variety of physical areas not specifically adapted for vehicle travel, a "road" may be a physical area not formally defined as a thoroughfare by any municipality or other governmental or administrative body.

[0044] As used herein, a "lane" is a portion of a road that can be traversed by a vehicle. A lane is sometimes identified based on lane markings. For example, a lane may correspond to most or all of the space between lane markings, or may correspond to only some (e.g., less than **50**%) of the space between lane markings. For example, a road having lane markings spaced far apart might accommodate two or more vehicles between the markings, such that one vehicle can pass the other without traversing the lane markings, and thus could be interpreted as having a lane narrower than the space between the lane markings, or having two lanes between the lane markings. A lane could also be interpreted in the absence of lane markings. For example, a lane may be defined based on physical features of an environment, e.g., rocks and trees along a thoroughfare in a rural area or, e.g., natural obstructions to be avoided in an undeveloped area. A lane could also be interpreted independent of lane markings or physical features. For example, a lane could be interpreted based on an arbitrary path free of obstructions in an area that otherwise lacks features that would be interpreted as lane boundaries. In an example scenario, an AV could interpret a lane through an obstruction-free portion of a field or empty lot. In another example scenario, an AV could interpret a lane through a wide (e.g., wide enough for two or more lanes) road that does not have lane markings. In this scenario, the AV could communicate information about the lane to other AVs so that the other AVs can use the same lane information to coordinate path planning among themselves.

[0045] The term "over-the-air (OTA) client" includes any AV, or any electronic device (e.g., computer, controller, IoT device, electronic control unit (ECU)) that is embedded in, coupled to, or in communication with an AV.

[0046] The term "over-the-air (OTA) update" means any update, change, deletion or addition to software, firmware, data or configuration settings, or any combination thereof, that is delivered to an OTA client using proprietary and/or standardized wireless communications technology, including but not limited to: cellular mobile communications (e.g., 2G, 3G, 4G, 5G), radio wireless area networks (e.g., WiFi) and/or satellite Internet.

[0047] The term "edge node" means one or more edge devices coupled to a network that provide a portal for communication with AVs and can communicate with other edge nodes and a cloud based computing platform, for scheduling and delivering OTA updates to OTA clients.

[0048] The term "edge device" means a device that implements an edge node and provides a physical wireless access point (AP) into enterprise or service provider (e.g., VERIZON, AT&T) core networks. Examples of edge devices include but are not limited to: computers, controllers, transmitters, routers, routing switches, integrated access devices (IADs), multiplexers, metropolitan area network (MAN) and wide area network (WAN) access devices.

[0049] "One or more" includes a function being performed by one element, a function being performed by more than one element, e.g., in a distributed fashion, several functions being performed by one element, several functions being performed by several elements, or any combination of the above.

[0050] It will also be understood that, although the terms first, second, etc. are, in some instances, used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first contact could be termed a second contact, and, similarly, a second contact could be termed a first contact, without departing from the scope of the various described embodiments. The first contact and the second contact are both contacts, but they are not the same contact.

[0051] The terminology used in the description of the various described embodiments herein is for the purpose of describing particular embodiments only and is not intended to be limiting. As used in the description of the various described embodiments and the appended claims, the singular forms "a," "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term "and/or" as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms "includes," "including," "comprises," and/or "comprising," when used in this description, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0052] As used herein, the term "if" is, optionally, construed to mean "when" or "upon" or "in response to determining" or "in response to detecting," depending on the context. Similarly, the phrase "if it is determined" or "if [a stated condition or event] is detected" is, optionally, construed to mean "upon determining" or "in response to determining" or "upon detecting [the stated condition or event]" or "in response to detecting [the stated condition or event]," depending on the context.

[0053] As used herein, an AV system refers to the AV along with the array of hardware, software, stored data, and data generated in real-time that supports the operation of the AV. In an embodiment, the AV system is incorporated within the AV. In an embodiment, the AV system is spread across several locations. For example, some of the software of the AV system is implemented in a cloud computing environment similar to cloud computing environment **200** described below with respect to FIG. **2**.

[0054] In general, this document describes technologies applicable to any vehicles that have one or more autonomous capabilities including fully autonomous vehicles, highly autonomous vehicles, and conditionally autonomous vehicles, such as so-called Level 5, Level 4 and Level 3 vehicles, respectively (see SAE International's standard J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, which is incorporated by reference in its entirety, for more details on the classification of levels of autonomy in vehicles). The technologies described in this document are also applicable to partially autonomous vehicles and driver assisted vehicles, such as so-called Level 2 and Level 1 vehicles (see SAE International's standard J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems). In an embodiment, one or more of the Level 1, 2, 3, 4 and 5 vehicle systems may automate certain vehicle operations (e.g., steering, braking, and using maps) under certain operating conditions based on processing of sensor inputs. The technologies described in this document can benefit vehicles in any levels, ranging from fully autonomous vehicles to human-operated vehicles.

[0055] Autonomous vehicles have advantages over vehicles that require a human driver. One advantage is safety. For example, in 2016, the United States experienced 6 million automobile accidents, 2.4 million injuries, 40,000 fatalities, and 13 million vehicles in crashes, estimated at a societal cost of $910+billion. U.S. traffic fatalities per 100 million miles traveled have been reduced from about six to about one from 1965 to 2015, in part due to additional safety measures deployed in vehicles. For example, an additional half second of warning that a crash is about to occur is believed to mitigate 60% of front-to-rear crashes. However, passive safety features (e.g., seat belts, airbags) have likely reached their limit in improving this number. Thus, active safety measures, such as automated control of a vehicle, are the likely next step in improving these statistics. Because human drivers are believed to be responsible for a critical pre-crash event in 95% of crashes, automated driving systems are likely to achieve better safety outcomes, e.g., by reliably recognizing and avoiding critical situations better than humans; making better decisions, obeying traffic laws, and predicting future events better than humans; and reliably controlling a vehicle better than a human.

[0056] Referring to FIG. **1**, an AV system **120** operates the AV **100** along a trajectory **198** through an environment **190** to a destination **199** (sometimes referred to as a final location) while avoiding objects (e.g., natural obstructions **191**, vehicles **193**, pedestrians **192**, cyclists, and other obstacles) and obeying rules of the road (e.g., rules of operation or driving preferences).

[0057] In an embodiment, the AV system **120** includes devices **101** that are instrumented to receive and act on operational commands from the computer processors **146**. We use the term "operational command" to mean an executable instruction (or set of instructions) that causes a vehicle to perform an action (e.g., a driving manuever). Operational commands can, without limitation, including instructions for a vehicle to start moving forward, stop moving forward, start moving backward, stop moving backward, accelerate, decelerate, perform a left turn, and perform a right turn. In an embodiment, are similar to the processor **304** described below in reference to FIG. **3**. Examples of devices **101** include a steering control **102**, brakes **103**, gears, accelerator pedal or other acceleration control mechanisms, windshield wipers, side-door locks, window controls, and turn-indicators.

[0058] In an embodiment, the AV system **120** includes sensors **121** for measuring or inferring properties of state or condition of the AV **100**, such as the AV's position, linear and angular velocity and acceleration, and heading (e.g., an orientation of the leading end of AV **100**). Example of sensors **121** are GPS, inertial measurement units (IMU) that measure both vehicle linear accelerations and angular rates, wheel speed sensors for measuring or estimating wheel slip ratios, wheel brake pressure or braking torque sensors, engine torque or wheel torque sensors, and steering angle and angular rate sensors.

[0059] In an embodiment, the sensors **121** also include sensors for sensing or measuring properties of the AV's environment. For example, monocular or stereo video cameras **122** in the visible light, infrared or thermal (or both) spectra, LiDAR 123, RADAR, ultrasonic sensors, time-offlight (TOF) depth sensors, speed sensors, temperature sensors, humidity sensors, and precipitation sensors.

[0060] In an embodiment, the AV system **120** includes a data storage unit **142** and memory **144** for storing machine instructions associated with computer processors **146** or data collected by sensors **121**. In an embodiment, the data storage unit **142** is similar to the ROM **308** or storage device **310** described below in relation to FIG. **3**. In an embodiment, memory **144** is similar to the main memory **306** described below. In an embodiment, the data storage unit **142** and memory **144** store historical, real-time, and/or predictive information about the environment **190**. In an embodiment, the stored information includes maps, driving performance, traffic congestion updates or weather conditions. In an embodiment, data relating to the environment **190** is transmitted to the AV **100** via a communications channel from a remotely located database **134**.

[0061] In an embodiment, the AV system **120** includes communications devices **140** for communicating measured or inferred properties of other vehicles' states and conditions, such as positions, linear and angular velocities, linear and angular accelerations, and linear and angular headings to the AV **100**. These devices include Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication devices and devices for wireless communications over point-to-point or ad hoc networks or both. In an embodiment, the communications devices **140** communicate across the electromagnetic spectrum (including radio and optical communications) or other media (e.g., air and acoustic media). A combination of Vehicle-to-Vehicle (V2V) Vehicle-to-Infrastructure (V2I) communication (and, in some embodiments, one or more other types of communication) is sometimes referred to as Vehicle-to-Everything (V2X) communication. V2X communication typically conforms to one or more communications standards for communication with, between, and among autonomous vehicles.

[0062] In an embodiment, the communication devices **140** include communication interfaces. For example, wired, wireless, WiMAX, Wi-Fi, Bluetooth, satellite, cellular, optical, near field, infrared, or radio interfaces. The communication interfaces transmit data from a remotely located database **134** to AV system **120**. In an embodiment, the remotely located database **134** is embedded in a cloud computing environment **200** as described in FIG. **2**. The communication interfaces **140** transmit data collected from sensors **121** or other data related to the operation of AV **100** to the remotely located database **134**. In an embodiment, communication interfaces **140** transmit information that relates to teleoperations to the AV **100**. In some embodiments, the AV **100** communicates with other remote (e.g., "cloud") servers **136**.

[0063] In an embodiment, the remotely located database **134** also stores and transmits digital data (e.g., storing data such as road and street locations). Such data is stored on the memory **144** on the AV **100**, or transmitted to the AV **100** via a communications channel from the remotely located database **134**.

[0064] In an embodiment, the remotely located database **134** stores and transmits historical information about driving properties (e.g., speed and acceleration profiles) of vehicles that have previously traveled along trajectory **198** at similar times of day. In one implementation, such data may be stored on the memory **144** on the AV **100**, or transmitted to the AV **100** via a communications channel from the remotely located database **134**.

[0065] Computing devices **146** located on the AV **100** algorithmically generate control actions based on both real-time sensor data and prior information, allowing the AV system **120** to execute its autonomous driving capabilities.

[0066] In an embodiment, the AV system **120** includes computer peripherals **132** coupled to computing devices **146** for providing information and alerts to, and receiving input from, a user (e.g., an occupant or a remote user) of the AV **100**. In an embodiment, peripherals **132** are similar to the display **312**, input device **314**, and cursor controller **316** discussed below in reference to FIG. **3**. The coupling is wireless or wired. Any two or more of the interface devices may be integrated into a single device.

[0067] In an embodiment, the AV system **120** receives and enforces a privacy level of a passenger, e.g., specified by the passenger or stored in a profile associated with the passenger. The privacy level of the passenger determines how particular information associated with the passenger (e.g., passenger comfort data, biometric data, etc.) is permitted to be used, stored in the passenger profile, and/or stored on the cloud server **136** and associated with the passenger profile. In an embodiment, the privacy level specifies particular information associated with a passenger that is deleted once the ride is completed. In an embodiment, the privacy level specifies particular information associated with a passenger and identifies one or more entities that are authorized to access the information. Examples of specified entities that are authorized to access information can include other AVs, third party AV systems, or any entity that could potentially access the information.

[0068] A privacy level of a passenger can be specified at one or more levels of granularity. In an embodiment, a privacy level identifies specific information to be stored or shared. In an embodiment, the privacy level applies to all the information associated with the passenger such that the passenger can specify that none of her personal information is stored or shared. Specification of the entities that are permitted to access particular information can also be specified at various levels of granularity. Various sets of entities that are permitted to access particular information can include, for example, other AVs, cloud servers **136**, specific third party AV systems, etc.

[0069] In an embodiment, the AV system **120** or the cloud server **136** determines if certain information associated with a passenger can be accessed by the AV **100** or another entity. For example, a third-party AV system that attempts to access passenger input related to a particular spatiotemporal location must obtain authorization, e.g., from the AV system **120** or the cloud server **136**, to access the information associated with the passenger. For example, the AV system **120** uses the passenger's specified privacy level to determine whether the passenger input related to the spatiotemporal location can be presented to the third-party AV system, the AV **100**, or to another AV. This enables the passenger's privacy level to specify which other entities are allowed to receive data about the passenger's actions or other data associated with the passenger.

[0070] FIG. **2** illustrates an example "cloud" computing environment. Cloud computing is a model of service delivery for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, network bandwidth, servers, processing, memory, storage, applications, virtual machines, and services). In typical cloud computing systems, one or more large cloud data centers house the machines used to deliver the services provided by the cloud. Referring now to FIG. **2**, the cloud computing environment **200** includes cloud data centers **204***a,* **204***b,* and **204***c* that are interconnected through the cloud **202**. Data centers **204***a,* **204***b,* and **204***c* provide cloud computing services to computer systems **206***a,* **206***b*, **206***c,* **206***d,* **206***e,* and **206***f* connected to cloud **202**.

[0071] The cloud computing environment **200** includes one or more cloud data centers. In general, a cloud data center, for example the cloud data center **204***a* shown in FIG. **2**, refers to the physical arrangement of servers that make up a cloud, for example the cloud **202** shown in FIG. **2**, or a particular portion of a cloud. For example, servers are physically arranged in the cloud datacenter into rooms, groups, rows, and racks. A cloud datacenter has one or more zones, which include one or more rooms of servers. Each room has one or more rows of servers, and each row includes one or more racks. Each rack includes one or more individual server nodes. In some implementation, servers in zones, rooms, racks, and/or rows are arranged into groups based on physical infrastructure requirements of the datacenter facility, which include power, energy, thermal, heat, and/or other requirements. In an embodiment, the server nodes are similar to the computer system described in FIG. **3**. The data center **204***a* has many computing systems distributed through many racks.

[0072] The cloud **202** includes cloud data centers **204***a,* **204***b,* and **204***c* along with the network and networking resources (for example, networking equipment, nodes, routers, switches, and networking cables) that interconnect the cloud data centers **204***a,* **204***b,* and **204***c* and help facilitate the computing systems' **206***a-f* access to cloud computing services. In an embodiment, the network represents any combination of one or more local networks, wide area networks, or internetworks coupled using wired or wireless links deployed using terrestrial or satellite connections. Data exchanged over the network, is transferred using any number of network layer protocols, such as Internet Protocol (IP), Multiprotocol Label Switching (MPLS), Asynchronous Transfer Mode (ATM), Frame Relay, etc. Furthermore, in embodiments where the network represents a combination of multiple sub-networks, different network layer protocols are used at each of the underlying sub-networks. In some embodiments, the network represents one or more interconnected internetworks, such as the public Internet.

[0073] The computing systems **206***a-f* or cloud computing services consumers are connected to the cloud **202** through network links and network adapters. In an embodiment, the computing systems **206***a-f* are implemented as various computing devices, for example servers, desktops, laptops, tablet, smartphones, Internet of Things (IoT) devices, autonomous vehicles (including, cars, drones, shuttles, trains, buses, etc.) and consumer electronics. In an embodiment, the computing systems **206***a-f* are implemented in or as a part of other systems.

[0074] FIG. **3** illustrates a computer system **300**. In an implementation, the computer system **300** is a special purpose computing device. The special-purpose computing device is hard-wired to perform the techniques or includes digital electronic devices such as one or more application-specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs) that are persistently programmed to perform the techniques, or may include one or more general purpose hardware processors programmed to perform the techniques pursuant to program instructions in firmware, memory, other storage, or a combination. Such special-purpose computing devices may also combine custom hard-wired logic, ASICs, or FPGAs with custom programming to accomplish the techniques. In various embodiments, the special-purpose computing devices are desktop computer systems, portable computer systems, handheld devices, network devices or any other device that incorporates hard-wired and/or program logic to implement the techniques.

[0075] In an embodiment, the computer system **300** includes a bus **302** or other communication mechanism for communicating information, and a hardware processor **304** coupled with a bus **302** for processing information. The hardware processor **304** is, for example, a general-purpose microprocessor. The computer system **300** also includes a main memory **306**, such as a random-access memory (RAM) or other dynamic storage device, coupled to the bus **302** for storing information and instructions to be executed by processor **304**. In one implementation, the main memory **306** is used for storing temporary variables or other intermediate information during execution of instructions to be executed by the processor **304**. Such instructions, when stored in non-transitory storage media accessible to the processor **304**, render the computer system **300** into a special-purpose machine that is customized to perform the operations specified in the instructions.

[0076] In an embodiment, the computer system **300** further includes a read only memory (ROM) **308** or other static storage device coupled to the bus **302** for storing static information and instructions for the processor **304**. A storage device **310**, such as a magnetic disk, optical disk, solid-state drive, or three-dimensional cross point memory is provided and coupled to the bus **302** for storing information and instructions.

[0077] In an embodiment, the computer system **300** is coupled via the bus **302** to a display **312**, such as a cathode ray tube (CRT), a liquid crystal display (LCD), plasma display, light emitting diode (LED) display, or an organic light emitting diode (OLED) display for displaying information to a computer user. An input device **314**, including alphanumeric and other keys, is coupled to bus **302** for communicating information and command selections to the processor **304**. Another type of user input device is a cursor controller **316**, such as a mouse, a trackball, a touch-enabled display, or cursor direction keys for communicating direction information and command selections to the processor **304** and for controlling cursor movement on the display **312**. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x-axis) and a second axis (e.g., y-axis), that allows the device to specify positions in a plane.

[0078] According to one embodiment, the techniques herein are performed by the computer system **300** in response to the processor **304** executing one or more sequences of one or more instructions contained in the main memory **306**. Such instructions are read into the main memory **306** from another storage medium, such as the storage device **310**. Execution of the sequences of instructions contained in the main memory **306** causes the processor **304** to perform the process steps described herein. In alternative embodiments, hard-wired circuitry is used in place of or in combination with software instructions.

[0079] The term "storage media" as used herein refers to any non-transitory media that store data and/or instructions that cause a machine to operate in a specific fashion. Such storage media includes non-volatile media and/or volatile media. Non-volatile media includes, for example, optical disks, magnetic disks, solid-state drives, or three-dimensional cross point memory, such as the storage device **310**. Volatile media includes dynamic memory, such as the main memory **306**. Common forms of storage media include, for example, a floppy disk, a flexible disk, hard disk, solid-state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium, any physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, NV-RAM, or any other memory chip or cartridge.

[0080] Storage media is distinct from but may be used in conjunction with transmission media. Transmission media participates in transferring information between storage media. For example, transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise the bus **302**. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infrared data communications.

[0081] In an embodiment, various forms of media are involved in carrying one or more sequences of one or more instructions to the processor **304** for execution. For example, the instructions are initially carried on a magnetic disk or solid-state drive of a remote computer. The remote computer loads the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to the computer system **300** receives the data on the telephone line and use an infrared transmitter to convert the data to an infrared signal. An infrared detector receives the data carried in the infrared signal and appropriate circuitry places the data on the bus **302**. The bus **302** carries the data to the main memory **306**, from which processor **304** retrieves and executes the instructions. The instructions received by the main memory **306** may optionally be stored on the storage device **310** either before or after execution by processor **304**.

[0082] The computer system **300** also includes a communication interface **318** coupled to the bus **302**. The communication interface **318** provides a two-way data communication coupling to a network link **320** that is connected to a local network **322**. For example, the communication interface **318** is an integrated service digital network (ISDN) card, cable modem, satellite modem, or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, the communication interface **318** is a local area network (LAN) card to provide a data communication connection to a compatible LAN. In some implementations, wireless links are also implemented. In any such implementation, the communication interface **318** sends and receives electrical, electromagnetic, or optical signals that carry digital data streams representing various types of information.

[0083] The network link **320** typically provides data communication through one or more networks to other data devices. For example, the network link **320** provides a connection through the local network **322** to a host computer **324** or to a cloud data center or equipment operated by an Internet Service Provider (ISP) **326**. The ISP **326** in turn provides data communication services through the worldwide packet data communication network now commonly referred to as the "Internet" **328**. The local network **322** and Internet **328** both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on the network link **320** and through the communication interface **318**, which carry the digital data to and from the computer system **300**, are example forms of transmission media. In an embodiment, the network **320** contains the cloud **202** or a part of the cloud **202** described above.

[0084] The computer system **300** sends messages and receives data, including program code, through the network(s), the network link **320**, and the communication interface **318**. In an embodiment, the computer system **300** receives code for processing. The received code is executed by the processor **304** as it is received, and/or stored in storage device **310**, or other non-volatile storage for later execution.

Autonomous Vehicle Architecture

[0085] FIG. **4** shows an example architecture **400** for an autonomous vehicle (e.g., the AV **100** shown in FIG. **1**). The architecture **400** includes a perception module **402** (sometimes referred to as a perception circuit), a planning module **404** (sometimes referred to as a planning circuit), a control module **406** (sometimes referred to as a control circuit), a localization module **408** (sometimes referred to as a localization circuit), and a database module **410** (sometimes referred to as a database circuit). Each module plays a role in the operation of the AV **100**. Together, the modules **402**, **404**, **406**, **408**, and **410** may be part of the AV system **120** shown in FIG. **1**. In some embodiments, any of the modules **402**, **404**, **406**, **408**, and **410** is a combination of computer software (e.g., executable code stored on a computer-readable medium) and computer hardware (e.g., one or more microprocessors, microcontrollers, application-specific integrated circuits [ASICs]), hardware memory devices, other types of integrated circuits, other types of computer hardware, or a combination of any or all of these things). Each of the modules **402**, **404**, **406**, **408**, and **410** is sometimes referred to as a processing circuit (e.g., computer hardware, computer software, or a combination of the two). A combination of any or all of the modules **402**, **404**, **406**, **408**, and **410** is also an example of a processing circuit.

[0086] In use, the planning module **404** receives data representing a destination **412** and determines data representing a trajectory **414** (sometimes referred to as a route) that can be traveled by the AV **100** to reach (e.g., arrive at) the destination **412**. In order for the planning module **404** to determine the data representing the trajectory **414**, the planning module **404** receives data from the perception module **402**, the localization module **408**, and the database module **410**.

[0087] The perception module **402** identifies nearby physical objects using one or more sensors **121**, e.g., as also shown in FIG. **1**. The objects are classified (e.g., grouped into types such as pedestrian, bicycle, automobile, traffic sign, etc.) and a scene description including the classified objects **416** is provided to the planning module **404**.

[0088] The planning module **404** also receives data representing the AV position **418** from the localization module **408**. The localization module **408** determines the AV position by using data from the sensors **121** and data from the database module **410** (e.g., a geographic data) to calculate a position. For example, the localization module **408** uses data from a GNSS (Global Navigation Satellite System) sensor and geographic data to calculate a longitude and latitude of the AV. In an embodiment, data used by the localization module **408** includes high-precision maps of the roadway geometric properties, maps describing road network connectivity properties, maps describing roadway physical properties (such as traffic speed, traffic volume, the number of vehicular and cyclist traffic lanes, lane width, lane traffic directions, or lane marker types and locations, or combinations of them), and maps describing the spatial locations of road features such as crosswalks, traffic signs or other travel signals of various types. In an embodiment, the high-precision maps are constructed by adding data through automatic or manual annotation to low-precision maps.

[0089] The control module **406** receives the data representing the trajectory **414** and the data representing the AV position **418** and operates the control functions **420**a-c (e.g., steering, throttling, braking, ignition) of the AV in a manner that will cause the AV **100** to travel the trajectory **414** to the destination **412**. For example, if the trajectory **414** includes a left turn, the control module **406** will operate the control functions **420**a-c in a manner such that the steering angle of the steering function will cause the AV **100** to turn left and the throttling and braking will cause the AV **100** to pause and wait for passing pedestrians or vehicles before the turn is made.

Autonomous Vehicle Inputs

[0090] FIG. **5** shows an example of inputs **502***a-d* (e.g., sensors **121** shown in FIG. **1**) and outputs **504***a-d* (e.g., sensor data) that is used by the perception module **402** (FIG. **4**). One input **502***a* is a LiDAR (Light Detection and Ranging) system (e.g., LiDAR **123** shown in FIG. **1**). LiDAR is a technology that uses light (e.g., bursts of light such as infrared light) to obtain data about physical objects in its line of sight. A LiDAR system produces LiDAR data as output **504***a*. For example, LiDAR data is collections of 3D or 2D points (also known as a point clouds) that are used to construct a representation of the environment **190**.

[0091] Another input **502***b* is a RADAR system. RADAR is a technology that uses radio waves to obtain data about nearby physical objects. RADARs can obtain data about objects not within the line of sight of a LiDAR system. A RADAR system **502***b* produces RADAR data as output **504***b*. For example, RADAR data are one or more radio frequency electromagnetic signals that are used to construct a representation of the environment **190**.

[0092] Another input **502***c* is a camera system. A camera system uses one or more cameras (e.g., digital cameras using a light sensor such as a charge-coupled device [CCD]) to obtain information about nearby physical objects. A camera system produces camera data as output **504***c*. Camera data often takes the form of image data (e.g., data in an image data format such as RAW, JPEG, PNG, etc.). In some examples, the camera system has multiple independent cameras, e.g., for the purpose of stereopsis (stereo vision), which enables the camera system to perceive depth. Although the objects perceived by the camera system are described here as "nearby," this is relative to the AV. In use, the camera system may be configured to "see" objects far, e.g., up to a kilometer or more ahead of the AV. Accordingly, the camera system may have features such as sensors and lenses that are optimized for perceiving objects that are far away.

[0093] Another input **502***d* is a traffic light detection (TLD) system. A TLD system uses one or more cameras to obtain information about traffic lights, street signs, and other physical objects that provide visual navigation information. A TLD system produces TLD data as output **504***d*. TLD data often takes the form of image data (e.g., data in an image data format such as RAW, JPEG, PNG, etc.). A TLD system differs from a system incorporating a camera in that a TLD system uses a camera with a wide field of view (e.g., using a wide-angle lens or a fish-eye lens) in order to obtain information about as many physical objects providing visual navigation information as possible, so that the AV **100** has access to all relevant navigation information provided by these objects. For example, the viewing angle of the TLD system may be about **120** degrees or more.

[0094] In some embodiments, outputs **504***a-d* are combined using a sensor fusion technique. Thus, either the individual outputs **504***a-d* are provided to other systems of the AV **100** (e.g., provided to a planning module **404** as shown in FIG. **4**), or the combined output can be provided to the other systems, either in the form of a single combined output or multiple combined outputs of the same type (e.g., using the same combination technique or combining the same outputs or both) or different types type (e.g., using different respective combination techniques or combining different respective outputs or both). In some embodiments, an early fusion technique is used. An early fusion technique is characterized by combining outputs before one or more data processing steps are applied to the combined output. In some embodiments, a late fusion technique is used. A late fusion technique is characterized by combining outputs after one or more data processing steps are applied to the individual outputs.

[0095] FIG. **6** shows an example of a LiDAR system **602** (e.g., the input **502***a* shown in FIG. **5**). The LiDAR system **602** emits light **604**a-c from a light emitter **606** (e.g., a laser transmitter). Light emitted by a LiDAR system is typically not in the visible spectrum; for example, infrared light is often used. Some of the light **604***b* emitted encounters a physical object **608** (e.g., a vehicle) and reflects back to the LiDAR system **602**. (Light emitted from a LiDAR system typically does not penetrate physical objects, e.g., physical objects in solid form.) The LiDAR system **602** also has one or more light detectors **610**, which detect the reflected light. In an embodiment, one or more data processing systems associated with the LiDAR system generates an image **612** representing the field of view **614** of the LiDAR system. The image **612** includes information that represents the boundaries **616** of a physical object **608**. In this way, the image **612** is used to determine the boundaries **616** of one or more physical objects near an AV.

[0096] FIG. **7** shows the LiDAR system **602** in operation. In the scenario shown in this figure, the AV **100** receives both camera system output **504***c* in the form of an image **702** and LiDAR system output **504***a* in the form of LiDAR data points **704**. In use, the data processing systems of the AV **100** compares the image **702** to the data points **704**. In particular, a physical object **706** identified in the image **702** is also identified among the data points **704**. In this way, the AV **100** perceives the boundaries of the physical object based on the contour and density of the data points **704**.

[0097] FIG. **8** shows the operation of the LiDAR system **602** in additional detail. As described above, the AV **100** detects the boundary of a physical object based on characteristics of the data points detected by the LiDAR system **602**. As shown in FIG. **8**, a flat object, such as the ground **802**, will reflect light **804**a-d emitted from a LiDAR system **602** in a consistent manner. Put another way, because the LiDAR system **602** emits light using consistent spacing, the ground **802** will reflect light back to the LiDAR system **602** with the same consistent spacing. As the AV **100** travels over the ground **802**, the LiDAR system **602** will continue to detect light reflected by the next valid ground point **806** if nothing is obstructing the road. However, if an object **808** obstructs the road, light **804***e-f* emitted by the LiDAR system **602** will be reflected from points **810***a-b* in a manner inconsistent with the expected consistent manner. From this information, the AV **100** can determine that the object **808** is present.

Path Planning

[0098] FIG. **9** shows a block diagram **900** of the relationships between inputs and outputs of a planning module **404** (e.g., as shown in FIG. **4**). In general, the output of a planning module **404** is a route **902** from a start point **904** (e.g., source location or initial location), and an end point **906** (e.g., destination or final location). The route **902** is typically defined by one or more segments. For example, a segment is a distance to be traveled over at least a portion of a street, road, highway, driveway, or other physical area appropriate for automobile travel. In some examples, e.g., if the AV **100** is an off-road capable vehicle such as a four-wheel-drive (4WD) or all-wheel-drive (AWD) car, SUV, pick-up truck, or the like, the route **902** includes "off-road" segments such as unpaved paths or open fields.

[0099] In addition to the route **902**, a planning module also outputs lane-level route planning data **908**. The lane-level route planning data **908** is used to traverse segments of the route **902** based on conditions of the segment at a particular time. For example, if the route **902** includes a multi-lane highway, the lane-level route planning data **908** includes trajectory planning data **910** that the AV **100** can use to choose a lane among the multiple lanes, e.g., based on whether an exit is approaching, whether one or more of the lanes have other vehicles, or other factors that vary over the course of a few minutes or less. Similarly, in some implementations, the lane-level route planning data **908** includes speed constraints **912** specific to a segment of the route **902**. For example, if the segment includes pedestrians or unexpected traffic, the speed constraints **912** may limit the AV **100** to a travel speed slower than an expected speed, e.g., a speed based on speed limit data for the segment.

[0100] In an embodiment, the inputs to the planning module **404** includes database data **914** (e.g., from the database module **410** shown in FIG. **4**), current location data **916** (e.g., the AV position **418** shown in FIG. **4**), destination data **918** (e.g., for the destination **412** shown in FIG. **4**), and object data **920** (e.g., the classified objects **416** as perceived by the perception module **402** as shown in FIG. **4**). In some embodiments, the database data **914** includes rules used in planning. Rules are specified using a formal language, e.g., using Boolean logic. In any given situation encountered by the AV **100**, at least some of the rules will apply to the situation. A rule applies to a given situation if the rule has conditions that are met based on information available to the AV **100**, e.g., information about the surrounding environment. Rules can have priority. For example, a rule that says, "if the road is a freeway, move to the leftmost lane" can have a lower priority than "if the exit is approaching within a mile, move to the rightmost lane."

[0101] FIG. **10** shows a directed graph **1000** used in path planning, e.g., by the planning module **404** (FIG. **4**). In general, a directed graph **1000** like the one shown in FIG. **10** is used to determine a path between any start point **1002** and end point **1004**. In real-world terms, the distance separating the start point **1002** and end point **1004** may be relatively large (e.g, in two different metropolitan areas) or may be relatively small (e.g., two intersections abutting a city block or two lanes of a multi-lane road).

[0102] In an embodiment, the directed graph **1000** has nodes **1006**a-d representing different locations between the start point **1002** and the end point **1004** that could be occupied by an AV **100**. In some examples, e.g., when the start point **1002** and end point **1004** represent different metropolitan areas, the nodes **1006**a-d represent segments of roads. In some examples, e.g., when the start point **1002** and the end point **1004** represent different locations on the same road, the nodes **1006**a-d represent different positions on that road. In this way, the directed graph **1000** includes information at varying levels of granularity. In an embodiment, a directed graph having high granularity is also a subgraph of another directed graph having a larger scale. For example, a directed graph in which the start point **1002** and the end point **1004** are far away (e.g., many miles apart) has most of its information at a low granularity and is based on stored data, but also includes some high granularity information for the portion of the graph that represents physical locations in the field of view of the AV **100**.

[0103] The nodes **1006**a-d are distinct from objects **1008***a-b* which cannot overlap with a node. In an embodiment, when granularity is low, the objects **1008***a-b* represent regions that cannot be traversed by automobile, e.g., areas that have no streets or roads. When granularity is high, the objects **1008***a-b* represent physical objects in the field of view of the AV **100**, e.g., other automobiles, pedestrians, or other entities with which the AV **100** cannot share physical space. In an embodiment, some or all of the objects **1008***a-b* are a static objects (e.g., an object that does not change position such as a street lamp or utility pole) or dynamic objects (e.g., an object that is capable of changing position such as a pedestrian or other car).

[0104] The nodes **1006***a-d* are connected by edges **1010***a-c*. If two nodes **1006***a-b* are connected by an edge **1010***a*, it is possible for an AV **100** to travel between one node **1006***a* and the other node **1006***b*, e.g., without having to travel to an intermediate node before arriving at the other node **1006***b*. (When we refer to an AV **100** traveling between nodes, we mean that the AV **100** travels between the two physical positions represented by the respective nodes.) The edges **1010***a-c* are often bidirectional, in the sense that an AV **100** travels from a first node to a second node, or from the second node to the first node. In an embodiment, edges **1010***a-c* are unidirectional, in the sense that an AV **100** can travel from a first node to a second node, however the AV **100** cannot travel from the second node to the first node. Edges **1010***a-c* are unidirectional when they represent, for example, one-way streets, individual lanes of a street, road, or highway, or other features that can only be traversed in one direction due to legal or physical constraints.

[0105] In an embodiment, the planning module **404** uses the directed graph **1000** to identify a path **1012** made up of nodes and edges between the start point **1002** and end point **1004**.

[0106] An edge **1010***a-c* has an associated cost **1014***a-b*. The cost **1014***a-b* is a value that represents the resources that will be expended if the AV **100** chooses that edge. A typical resource is time. For example, if one edge **1010***a* represents a physical distance that is twice that as another edge **1010***b*, then the associated cost **1014***a* of the first edge **1010***a* may be twice the associated cost **1014***b* of the second edge **1010***b*. Other factors that affect time include expected traffic, number of intersections, speed limit, etc. Another typical resource is fuel economy. Two edges **1010***a-b* may represent the same physical distance, but one edge **1010***a* may require more fuel than another edge **1010***b*, e.g., because of road conditions, expected weather, etc.

[0107] When the planning module **404** identifies a path **1012** between the start point **1002** and end point **1004**, the planning module **404** typically chooses a path optimized for cost, e.g., the path that has the least total cost when the individual costs of the edges are added together.

Autonomous Vehicle Control

[0108] FIG. **11** shows a block diagram **1100** of the inputs and outputs of a control module **406** (e.g., as shown in FIG. **4**). A control module operates in accordance with a controller **1102** which includes, for example, one or more processors (e.g., one or more computer processors such as microprocessors or microcontrollers or both) similar to processor **304**, short-term and/or long-term data storage (e.g., memory random-access memory or flash memory or both) similar to main memory **306**, ROM **308**, and storage device **310**, and instructions stored in memory that carry out operations of the controller **1102** when the instructions are executed (e.g., by the one or more processors).

[0109] In an embodiment, the controller **1102** receives data representing a desired output **1104**. The desired output **1104** typically includes a velocity, e.g., a speed and a heading. The desired output **1104** can be based on, for example, data received from a planning module **404** (e.g., as shown in FIG. **4**). In accordance with the desired output **1104**, the controller **1102** produces data usable as a throttle input **1106** and a steering input **1108**. The throttle input **1106** represents the magnitude in which to engage the throttle (e.g., acceleration control) of an AV **100**, e.g., by engaging the steering pedal, or engaging another throttle control, to achieve the desired output **1104**. In some examples, the throttle input **1106** also includes data usable to engage the brake (e.g., deceleration control) of the AV **100**. The steering input **1108** represents a steering angle, e.g., the angle at which the steering control (e.g., steering wheel, steering angle actuator, or other functionality for controlling steering angle) of the AV should be positioned to achieve the desired output **1104**.

[0110] In an embodiment, the controller **1102** receives feedback that is used in adjusting the inputs provided to the throttle and steering. For example, if the AV **100** encounters a disturbance **1110**, such as a hill, the measured speed **1112** of the AV **100** is lowered below the desired output speed. In an embodiment, any measured output **1114** is provided to the controller **1102** so that the necessary adjustments are performed, e.g., based on the differential **1113** between the measured speed and desired output. The measured output **1114** includes measured position **1116**, measured velocity **1118**, (including speed and heading), measured acceleration **1120**, and other outputs measurable by sensors of the AV **100**.

[0111] In an embodiment, information about the disturbance **1110** is detected in advance, e.g., by a sensor such as a camera or LiDAR sensor, and provided to a predictive feedback module **1122**. The predictive feedback module **1122** then provides information to the controller **1102** that the controller **1102** can use to adjust accordingly. For example, if the sensors of the AV **100** detect ("see") a hill, this information can be used by the controller **1102** to prepare to engage the throttle at the appropriate time to avoid significant deceleration.

[0112] FIG. **12** shows a block diagram **1200** of the inputs, outputs, and components of the controller **1102**. The controller **1102** has a speed profiler **1202** which affects the operation of a throttle/brake controller **1204**. For example, the speed profiler **1202** instructs the throttle/brake controller **1204** to engage acceleration or engage deceleration using the throttle/brake **1206** depending on, e.g., feedback received by the controller **1102** and processed by the speed profiler **1202**.

[0113] The controller **1102** also has a lateral tracking controller **1208** which affects the operation of a steering controller **1210**. For example, the lateral tracking controller **1208** instructs the steering controller **1210** to adjust the position of the steering angle actuator **1212** depending on, e.g., feedback received by the controller **1102** and processed by the lateral tracking controller **1208**.

[0114] The controller **1102** receives several inputs used to determine how to control the throttle/brake **1206** and steering angle actuator **1212**. A planning module **404** provides information used by the controller **1102**, for example, to choose a heading when the AV **100** begins operation and to determine which road segment to traverse when the AV **100** reaches an intersection. A localization module **408** provides information to the controller **1102** describing the current location of the AV **100**, for example, so that the controller **1102** can determine if the AV **100** is at a location expected based on the manner in which the throttle/brake **1206** and steering angle actuator **1212** are being controlled. In an embodiment, the controller **1102** receives information from other inputs **1214**, e.g., information received from databases, computer networks, etc.

Background Detection

[0115] Urban areas are cluttered. Dynamic entities such as vehicles with various sizes and shapes, pedestrians with non-rigid motions, rapidly changing object occlusion, construction zones, trees and foliage are just a few factors for the clutter. Dynamic entities further include objects (e.g., natural obstructions **191**, vehicles **193**, pedestrians **192**, cyclists, and other obstacles). In some cases, this often renders detection of object boundaries, hence segmentation, from point cloud data collected with LiDAR(s) difficult due to the sparsity of the data. Consecutive measurements taken from an object with non-zero relative motion with respect to the sensor often do not overlap sufficiently prohibiting temporal data association. On the other hand, LiDAR devices are more accurate at measuring distance than cameras. A sparse point cloud processing pipeline can be used to leverage the accuracy of LiDAR in its early stages to segment out the ground surface. This facilitates foreground object segmentation and their temporal association in the later stages of the pipeline.

[0116] In an example, a ground surface exhibits properties that are easy to model compared to other types of objects. Ground planes also virtually connect all objects to others since almost all objects in an urban environment sit on the ground surface. In turn, this means that segmenting out the ground surface disconnects most objects, facilitating drawing object boundaries. Due to these two reasons, the first stage of a point cloud processing pipeline labels ground surface points.

[0117] Ground surface is typically assumed to be planar, but this assumption does not always hold in the real world (e.g., hilly environments such as San Francisco). In some examples, region growing can be used based on metrics such as point/facet normal, point-to-point distance, etc., starting from randomly selected seed points making them prone to failure in case of a bad seed point selection.

[0118] Unlike camera images, LiDAR point clouds do not inherently contain point (pixel) neighborhood information. Although a driver (e.g., interface) of a LiDAR can be configured to publish ordered point cloud data, neighborhood relations are still reconstructed for multi-LiDAR scenarios, e.g., autonomous vehicles. Two methods to implement a for neighborhood search are k-nearest-neighbor (kNN) search and radius-search (r-search). These methods work well on uniformly sampled point clouds. However, due to the projective nature of the LiDAR sensors, point-to-point distance exponentially increases with range for data collected from the ground surface. In some examples the two search methods are run with different parameters adaptively changing at different ranges.

[0119] A 2D mesh of the environment can be used. For this, all the points are projected on a unit sphere centered at the (virtual) sensor origin and 2D Delaunay triangulation is performed on the sphere. These points are then back-projected to their original ranges to give the final triangular mesh. Spherical rotary tessellation is used to generate "faces"—planes between points in a point cloud. Generally, tessellation of a sphere covers the sphere in tiles. In an embodiment, the tiles are triangular, without overlapping tiles or gaps.

[0120] Spherical Delaunay triangulation can be used with a reformulation of Fortune's sweep-line algorithm, which is originally designed for a 2D plane. Generally, a Delaunay triangulation for a given set of discrete points in a general position is a triangulation such that no point in is inside the circumcircle of any triangle in of the triangulation. In an embodiment, the Delaunay triangulation maximizes the minimum angle of all the angles of the triangles in the triangulation and tends to avoid sliver triangles. In examples, the sweep-circle is initialized according to the a priori ground plane direction which is estimated by a Kalman Filter that tracks the gravity vector using an onboard Inertial Measurement Unit (IMU). A starting position of the sweep-circle determines the evolution of the Delaunay triangulation. For example, in a point cloud collected with a LiDAR oriented such that its -z axis coincides with the gravity vector, the meshing will start adding triangles with vertices at higher z coordinates. However, parts of a mesh with higher z coordinates may not correspond to the ground surface, e.g., building facades or tree canopies. Thus, the point cloud is transformed such that the meshing starts off from the points that are more likely to be corresponding to the ground surface. In embodiments, meshing begins with ground surface points. In this way, not all of the points need to be processed since the ground surface resides on the north hemisphere of the sweep-sphere (e.g., after transformation). In some examples, this saves CPU time. Further, due to the projective nature of a LiDAR sensor and how the meshing works, many sliver, slant artificial faces are added to the mesh. Given that the (virtual, if multiple LiDARs are used) sensor origin is selected properly, such faces are always due to non-ground objects. By initializing the sweep-circle at the proper point, addition of such faces into the mesh can be avoided, which might otherwise cause excess noise obstructing the ground surface detection. In embodiments, the proper point is a point that is likely a ground surface point based on the estimated ground plane direction.

[0121] Together with the mesh of the environment, a graph with its vertices corresponding to the mesh faces is constructed concurrently. Graph edges are defined only between faces that are similar in orientation and elevation. The mesh is then over-segmented by performing connected component analysis on this graph. Connected components with more than a threshold number of elements, and those with their mean surface normal close to the a priori ground plane normal, are identified as the ground surface. An alternative approach to the connected component analysis is to perform graph cut, though that technique may use more processing power.

[0122] This technique uses few parameters such as maximum surface normal deviation and connected component size. This technique also does not use any assumptions on the planarity of the ground surface. Further, multiple LiDARS can be used because during the meshing process, neighborhood relations are constructed by the algorithm. For example, a point is associated with other points according to the identified faces as described below.

[0123] After the ground surface points are removed, the later stages of a point cloud processing pipeline can more easily detect objects such as cars and pedestrians since each such object will be isolated from its surroundings, making it possible to perform segmentation using simple Euclidean distance.

[0124] FIG. **13** shows a partial mesh from a raw point cloud. In an embodiment, the present techniques enable receiving line or point cloud information from the at least one LiDAR device. In the example of FIG. **13**, generally, a mesh is a structure applied to a plurality of data points. A LiDAR, such as LiDAR **123** of FIG. **1** outputs a collection of two dimensional (2D) or three dimensional (3D) points **1302**. In some cases, the 2D or 3D points **1302** are referred to as point clouds (e.g., output **504***a* of FIG. **5**) that are used to construct a representation of the environment **190** (FIG. **1**).

[0125] In an embodiment, the partial mesh **1300** is line or point cloud information received from at least one LiDAR device. The partial mesh is a two dimensional (2D) representation of point cloud data points in the environment. The present techniques are operable using one or more LiDAR devices. In an example with a plurality of LiDAR devices, the sensor origin is selected properly when modeling the point cloud information as a sphere. In embodiments, after obtaining point cloud information from at least one LiDAR device, the point cloud information is modeled as a sphere.

[0126] Accordingly, the present techniques enable modeling the point cloud information as a sphere. In an embodiment, to model the point cloud information as a sphere, a two-dimensional mesh of the environment is generated using the point cloud data. In an example, to generate the 2D mesh of the environment, the point cloud points are projected onto a unit sphere centered at the (virtual) sensor origin. Generally, point cloud data has an associated set of X, Y, and Z, coordinates. Projecting the point cloud data onto the unit sphere maps the point cloud data to a set of spherical coordinates. In an embodiment, the spherical coordinates specify a position of a point by three numbers: the radial distance of that point from a fixed origin, its polar angle measured from a fixed zenith direction, and the azimuthal angle of its orthogonal projection on a reference plane that passes through the origin and is orthogonal to the zenith, measured from a fixed reference direction on that plane. For example, a spherical coordinate system can be realized as the three-dimensional version of the polar coordinate system. In embodiments, to model the point cloud information is a sphere, the point cloud is managed using spherical parameterization. In an embodiment, clusters on the sphere are used to identify faces in the point cloud data as described herein.

[0127] In an embodiment, two-dimensional triangulation is performed on the unit sphere to generate the polygonal mesh, also referred to as meshing. In an embodiment the output of spherical rotary tessellation is one or more faces. As used herein, a face is a plane between points in a point cloud. In an embodiment, the faces may be generated using spherical Delaunay triangulation with a reformulation of fortune's line sweep algorithm. Fortunes line sweep algorithm was originally designed for use with a two dimensional plane. The line sweep (and meshing) begins from the points that are more likely to correspond to the ground surface. Generally, after transformation from a point cloud to a sphere, the ground surface resides on the north hemisphere of the sweep-sphere.

[0128] FIG. **14** shows faces representing the ground surface **1402** as well as other objects **1404**. Generally, the ground surface **1402** is represented by white triangles, while the other objects **1404** are represented by filled/pattered triangles. In an embodiment, based on the sphere, faces are identified corresponding to respective clusters of points of the received LiDAR point cloud information. The faces **1402** represent a ground surface. The faces **1404** represent other objects. In an embodiment, lines are used to connect each site with its nearest neighbors. The triangulation used to form the meshing minimizes the degeneracy of the generated triangles.

[0129] In generating the mesh **1402**, the point cloud is transformed such that the meshing begins at points that are more likely to correspond to the ground surface. In this way, not all the points need to be processed since the ground surface resides on the North hemisphere of the sweep hemisphere. In some examples, this saves CPU time. Further, due to the projected nature of a LiDAR sensor and how the meshing works, many sliver, slant artificial faces are added to the mesh. If the sensor origin is selected properly, the addition of such spaces into the mesh can be avoided, which might otherwise cause excess noise obstructing the ground surface detection. In an embodiment, processing ends when the ground surface is identified.

[0130] FIG. **15** shows the identified ground surface **1402**.

[0131] FIG. **16** shows the identified non-ground faces **1404**.

[0132] FIG. **17** shows the correspondence **1700** of exemplary faces F1-F5 and a graph data structure. The faces F1-F5 may be, for example, faces **1402** or **1404**. For example, faces F2-F4 are adjacent (e.g., share LiDAR point cloud points, represented as triangles in the figure) and share similar characteristics and thus their corresponding graph nodes/vertices (represented as circles) belong to a connected graph. Faces F1 and F5, while adjacent to other faces, do not share characteristics (e.g., within a threshold of similarity) and thus do not connect to the other nodes/vertices.

[0133] In an embodiment, a graph structure **1702** is generated that includes vertices corresponding to the respective faces of the identified mesh faces. In an embodiment, together with the mesh of the environment, a graph with its vertices corresponding to the mesh faces is constructed concurrently (with the mesh of the environment). Graph edges are defined only between faces that are similar in orientation and elevation.

[0134] In an embodiment, the present techniques enable connecting the vertices of the graph data structure based on adjacency of the respective underlying points and characteristics of the corresponding faces. As used herein, faces that are adjacent share one or more point cloud points. In the graph, the faces F1-F5 are represented by nodes. In an embodiment, FIG. **17** is an illustration of the correspondence of faces F1 through F5 **1702** and a graph data structure **1704**. In the graph structure **1704** each face **1702** is represented by a node. Connections between the node/vertices of the graph may be based, at least in part, on the adjacency of the respective underlying points from the point cloud. Connections between the nodes/vertices of the graph may be based, at least in part, on characteristics of the corresponding faces. A characteristic, may be, for example that the faces are within a threshold of similarity. As used herein, a threshold of similarity refers to a similar orientation or elevation. For example, faces with a similar orientation are oriented in a substantially similar direction. Faces with a similar elevation are positioned at a substantially similar elevation.

[0135] In an embodiment, based on the graph data structure, a set of graphs is identified, wherein each subgraph includes connected vertices. Sub-graph generation is based on graph edges that are defined only between faces that are similar in orientation and elevation. In an embodiment, for each subgraph, characteristics of the faces corresponding to the vertices of the subgraph are analyzed. For example, the mesh is then over segmented by performing connected component analysis on this graph. Connected components with more than a threshold number of elements, and those with their mean surface normal close to the high priority ground plane normal, are identified as ground surface. An alternative approach to the connected component analysis is to perform graph cut, though that technique may use more processing power. In an embodiment, based on the analysis, the subgraph that corresponds to a background feature of the physical environment is identified.

[0136] FIG. **18** is a process flow diagram of a process for identifying background features using LiDAR. In the example of FIG. **18**, the process may be implemented by computing processors **146** of FIG. **1**. The process may be implemented in a cloud computing environment similar to cloud computing environment **200** described with respect to

FIG. **2**. Additionally, in the example of FIG. **18**, the process may be implemented by the computer system **300** of FIG. **3**.

[0137] At block **1802**, LiDAR point cloud information (e.g., output **504**a) is obtained from the at least one LiDAR device (e.g., LiDAR **123**). At block **1804**, the point cloud information is modeled as a sphere. At block **1806**, based on the sphere, faces corresponding to respective clusters of points of the received LiDAR point cloud information are identified. For example, spherical rotary tessellation is applied to the sphere to identify at least one face.

[0138] At block **1808**, a graph data structure is generated that includes vertices corresponding to respective faces of the identified faces. At block **1810**, vertices of the graph data structure are connected based on adjacency of the respective underlying points and characteristics of the corresponding faces. At block **1812**, based on the graph data structure, subgraphs are identified, wherein each subgraph includes connected vertices. At block **1814**, for each subgraph, characteristics of the faces corresponding to the vertices of the subgraph are analyzed. At block **1816**, based on the analysis, subgraphs are identified that correspond to a background feature of the physical environment. At block **1818**, a control circuit communicatively coupled to the at least one processor is configured to operate the vehicle based upon the identified background feature.

[0139] In the foregoing description, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation.

[0140] The description and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. The sole and exclusive indicator of the scope of the invention, and what is intended by the applicants to be the scope of the invention, is the literal and equivalent scope of the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions expressly set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. In addition, when we use the term "further comprising," in the foregoing description or following claims, what follows this phrase can be an additional step or entity, or a sub-step/sub-entity of a previously-recited step or entity.

1. A vehicle, comprising:

at least one LiDAR device configured to detect electromagnetic radiation reflected from objects proximate to the vehicle and generate LiDAR point cloud information based on the detected light;

at least one computer-readable medium storing computer-executable instructions;

at least one processor communicatively coupled to the at least one LiDAR device and configured to execute the computer executable instructions, the execution carrying out operations comprising:

receiving LiDAR point cloud information from the at least one LiDAR device;

modeling the point cloud information as a sphere;

based on the sphere, identifying faces corresponding to respective clusters of points of the received LiDAR point cloud information;

generating a graph data structure that includes vertices corresponding to respective faces of the identified faces;

connecting vertices of the graph data structure based on adjacency of the respective underlying points and characteristics of the corresponding faces;

based on the graph data structure, identifying subgraphs, wherein each subgraph includes connected vertices;

for each subgraph, analyzing characteristics of the faces corresponding to the vertices of the subgraph; and

based on the analysis, identifying subgraphs that correspond to a background feature of the physical environment; and

a control circuit communicatively coupled to the at least one processor, wherein the control circuit is configured to operate the vehicle based upon the identified background feature.

2. The vehicle of claim **1**, wherein the operations comprising modeling the point cloud information as a sphere includes projecting the point cloud onto a unit sphere.

3. The vehicle of claim **1**, wherein a sparsity associated with the point cloud information increases as distance from the LiDAR increases.

4. The vehicle of claim **1**, wherein the point could information is not uniformly sampled.

5. The vehicle of claim **1**, wherein the operations comprising identifying faces corresponding to respective clusters of points of the received LiDAR point cloud information comprises spherical rotary tessellation.

6. The vehicle of claim **1**, wherein a face is a plane between a plurality of points in the point cloud information.

7. The vehicle of claim **1**, wherein an inertial measurement unit is used to determine a gravity vector, wherein the gravity vector is used to estimate a ground plane direction for initializing a sweep circle of spherical Delaunay triangulation, wherein spherical Delaunay triangulation can be used with a reformulation of Fortune's sweep-line algorithm to identify faces.

8. The vehicle of claim **7**, wherein initializing the sweep-circle at a proper point enables the elimination of clutter into a mesh.

9. The vehicle of claim **1**, wherein the operations comprising identifying faces corresponding to respective clusters of points of the received LiDAR point cloud information comprises initializing meshing at points that correspond to the ground surface.

10. The vehicle of claim **1**, wherein a graph data structure and a mesh are constructed concurrently.

11. A method, comprising:

receiving, from at least one LiDAR device, LiDAR point cloud information;

modeling, using at least one processor, the point cloud information as a sphere;

identifying, using the at least one processor, faces corresponding to respective clusters of points of the received LiDAR point cloud information;

generating, using the at least one processor, a graph data structure that includes vertices corresponding to respective faces of the identified faces;

connecting, using the at least one processor, vertices of the graph data structure based on adjacency of the respective underlying points and characteristics of the corresponding faces;

identifying, using the at least one processor, subgraphs of the graph data structure, wherein each subgraph includes connected vertices;

analyzing, using the at least one processor, characteristics of the faces corresponding to the vertices of the subgraph; and

identifying, using the at least one processor, subgraphs that correspond to a background feature of the physical environment based on the analysis; and

operating, using a control circuit, the vehicle based upon the identified background feature.

12. The method of claim **11**, comprising projecting the point cloud onto a unit sphere.

13. The method of claim **11**, wherein a sparsity associated with the point cloud information increases as distance from the LiDAR increases.

14. The method of claim **11**, wherein the point could information is not uniformly sampled.

15. The method of claim **11**, comprising spherical rotary tessellation to identify faces corresponding to respective clusters of points of the received LiDAR point cloud information.

16. The method of claim **11**, wherein a face is a plane between a plurality of points in the point cloud information.

17. The method of claim **11**, wherein an inertial measurement unit is used to determine a gravity vector, wherein the gravity vector is used to estimate a ground plane direction for initializing a sweep circle of spherical Delaunay triangulation, wherein spherical Delaunay triangulation can be used with a reformulation of Fortune's sweep-line algorithm to identify faces.

18. The method of claim **17**, wherein initializing the sweep-circle at a proper point enables the elimination of clutter into a mesh.

19. A non-transitory computer-readable storage medium comprising at least one program for execution by at least one processor of a first device, the at least one program including instructions which, when executed by the at least one processor, carry out a method comprising:

receiving, from at least one LiDAR device, LiDAR point cloud information;

modeling, using at least one processor, the point cloud information as a sphere;

identifying, using the at least one processor, faces corresponding to respective clusters of points of the received LiDAR point cloud information;

generating, using the at least one processor, a graph data structure that includes vertices corresponding to respective faces of the identified faces;

connecting, using the at least one processor, vertices of the graph data structure based on adjacency of the respective underlying points and characteristics of the corresponding faces;

identifying, using the at least one processor, subgraphs of the graph data structure, wherein each subgraph includes connected vertices;

analyzing, using the at least one processor, characteristics of the faces corresponding to the vertices of the subgraph; and

identifying, using the at least one processor, subgraphs that correspond to a background feature of the physical environment based on the analysis; and

operating, using a control circuit, the vehicle based upon the identified background feature.

20. The non-transitory computer-readable storage medium of claim **19**, comprising projecting the point cloud onto a unit sphere.

* * * * *