(19) Deutsches Patent- und Markenamt

(10) **DE 10 2022 126 707 A1** 2023.05.04

(12) # Offenlegungsschrift

(21) Aktenzeichen: **10 2022 126 707.5**
(22) Anmeldetag: **13.10.2022**
(43) Offenlegungstag: **04.05.2023**

(51) Int Cl.: ***G06T 7/521*** (2017.01)
G06N 3/02 (2006.01)

(30) Unionspriorität:
17/452,752 **28.10.2021** **US**

(71) Anmelder:
NVIDIA Corporation, Santa Clara, CA, US

(74) Vertreter:
Kraus & Weisert Patentanwälte PartGmbB, 80539 München, DE

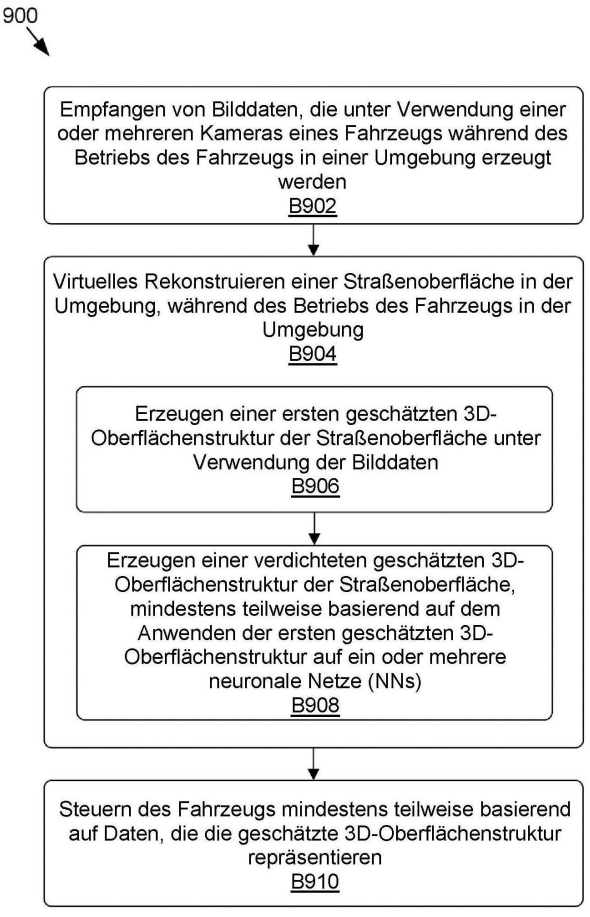
(72) Erfinder:
Wang, Kang, Santa Clara, CA, US; Wu, Yue, Santa Clara, CA, US; Park, Minwoo, Santa Clara, CA, US; Pan, Gang, Santa Clara, CA, US

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Verwendung neuronaler Netze zur 3D-Oberflächenstrukturschätzung basierend auf realen Daten für autonome Systeme und Anwendungen**

(57) Zusammenfassung: In verschiedenen Beispielen wird zur Unterstützung des Trainings eines tiefen neuronalen Netzes (DNN) zur Vorhersage einer dichten Darstellung einer interessierenden 3D-Oberflächenstruktur ein Trainingsdatensatz aus realen Daten erzeugt. Zum Beispiel können ein oder mehrere Fahrzeuge Bilddaten und LiDAR-Daten sammeln, während sie durch eine reale Umgebung navigieren. Um Eingabetrainingsdaten zu erzeugen, kann eine 3D-Oberflächenstrukturschätzung an den erfassten Bilddaten durchgeführt werden, um eine spärliche Darstellung einer interessierenden 3D-Oberflächenstruktur (z. B. einer 3D-Straßenoberfläche) zu erzeugen. Um entsprechende Ground-Truth-Trainingsdaten zu erzeugen, können die erfassten LiDAR-Daten geglättet, einer Ausreißerentfernung unterzogen, einer Triangulation unterzogen werden, um fehlende Werte aufzufüllen, von mehreren LiDAR-Sensoren akkumuliert, mit entsprechenden Frames von Bilddaten ausgerichtet und/oder annotiert werden, um 3D-Punkte auf der interessierenden 3D-Oberfläche zu identifizieren, und die identifizierten 3D-Punkte können projiziert werden, um eine dichte Darstellung der 3D-Oberflächenstruktur zu erzeugen.



Beschreibung

Hintergrund

[0001] Das Entwerfen eines Systems zum autonomen, sicheren und komfortablen Fahren eines Fahrzeugs ohne Aufsicht ist enorm schwierig. Ein autonomes Fahrzeug sollte mindestens in der Lage sein, als funktionales Äquivalent eines aufmerksamen Fahrers zu fungieren - der auf ein Wahrnehmungs- und Handlungssystem zurückgreift, das eine unglaubliche Fähigkeit besitzt, bewegliche und statische Hindernisse in einer komplexen Umgebung zu erkennen und darauf zu reagieren - um entlang des Wegs des Fahrzeugs durch die umgebende dreidimensionale (3D) Umgebung zu navigieren. Daher ist die Fähigkeit, Teile einer Umgebung zu erkennen, oft entscheidend für Wahrnehmungssysteme beim autonomen Fahren. Diese Fähigkeit wird immer wichtiger, da die Betriebsumgebung für das autonome Fahrzeug begonnen hat, sich von Autobahnumgebungen auf halbstädtische und städtische Umgebungen auszudehnen, die durch komplexe Szenen mit komplexen Formen gekennzeichnet sind.

[0002] Eine wichtige Komponente der 3D-Umgebung ist die 3D-Straßenoberfläche. Die Kenntnis der 3D-Straßenoberfläche ermöglicht es autonomen Fahrzeugen, ein komfortables und sicheres Fahrerlebnis zu bieten. So kann ein autonomes Fahrzeug beispielsweise die Aufhängung des Fahrzeugs an die aktuelle Straßenoberfläche anpassen (z. B. durch Kompensation von Unebenheiten auf der Straße). In einem anderen Beispiel kann ein autonomes Fahrzeug so navigieren, dass es Unebenheiten (z. B. Vertiefungen, Löcher) in der Fahrbahn vermeidet. In einem weiteren Beispiel kann ein autonomes Fahrzeug eine frühzeitige Beschleunigung oder Abbremsung basierend auf einer sich nähernden Neigung auf der Straße vornehmen. Jede dieser Funktionen kann dazu dienen, die Sicherheit zu erhöhen, die Langlebigkeit des Fahrzeugs zu verbessern, die Energieeffizienz zu steigern und/oder ein angenehmes Fahrgefühl zu vermitteln.

[0003] Eine Art, die Struktur der Straßenoberfläche abzuschätzen, ist die 3D-Rekonstruktion. Bestehende Ansätze zur 3D-Rekonstruktion der Straßenoberfläche stützen sich entweder auf LiDAR-Sensoren oder Kameras. Herkömmliche Verfahren, die LiDAR-Sensoren verwenden, senden einen Laserimpuls aus und erfassen das von der Straßenoberfläche reflektierte Signal, um 3D-Punkte auf der Straße zu rekonstruieren. LiDAR-Sensoren sind jedoch teuer, haben eine begrenzte Reichweite und ihre Genauigkeit reicht für bestimmte Anwendungen beim autonomen Fahren möglicherweise nicht aus. Herkömmliche Techniken, die Kameras verwenden, stützen sich auf die Geometrie mehrerer Ansichten, um 3D-Elemente zu rekonstruieren. Herkömmliche Rekonstruktionstechniken mit Kameras können jedoch keine dichten Messungen effizient berechnen, und herkömmliche Nachbearbeitungstechniken wie Interpolation oder Ebenenanpassung reichen oft nicht aus, um hinreichend genaue Modelle der komplexen Straßenoberflächen zu erstellen, die in der realen Welt existieren. Daher besteht ein Bedarf an verbesserten 3D-Straßenoberflächenrekonstruktionsverfahren für autonome Fahranwendungen.

Zusammenfassung

[0004] Ausführungsformen der vorliegenden Offenbarung betreffen die 3D-Oberflächenschätzung. In einigen Ausführungsformen kann eine 3D-Oberflächenstruktur wie die 3D-Oberflächenstruktur einer Straße (3D-Straßenoberfläche) beobachtet und geschätzt werden, um eine 3D-Punktwolke oder eine andere Darstellung der 3D-Oberflächenstruktur zu erzeugen. Da die Darstellung spärlich sein kann, können eine oder mehrere Verdichtungstechniken angewendet werden, um eine dichte Darstellung der 3D-Oberflächenstruktur zu erzeugen, die einem autonomen Fahrzeugfahrstapel zur Verfügung gestellt werden kann, um eine sichere und komfortable Planung und Steuerung des autonomen Fahrzeugs zu ermöglichen.

[0005] In einer beispielhaften Ausführungsform können eine oder mehrere Kameras an einem Fahrzeug oder einem anderen Objekt befestigt oder anderweitig angeordnet sein und dazu verwendet werden, Bilder einer 3D-Umgebung aufzunehmen, während das Fahrzeug oder Objekt (z. B. entlang einer Straße) durch die 3D-Umgebung navigiert, und jede geeignete 3D-Strukturschätzungstechnik kann angewendet werden, um eine Darstellung einer interessierenden 3D-Oberflächenstruktur, wie z. B. einer 3D-Straßenoberfläche, zu erzeugen. Die Darstellung der 3D-Oberflächenstruktur kann beispielsweise mithilfe eines Markov-Zufallsfeldes und/oder eines tiefen neuronalen Netzes (DNN) verdichtet werden. Bei einer beispielhaften Verdichtungstechnik, die ein Markov-Zufallsfeld verwendet, können spärliche und dichte Projektionsbilder (z. B. Höhenkarten) mit einem ungerichteten Graphen modelliert werden, und die Maximum-a-Posterior (MAP)-Inferenz kann verwendet werden, um die wahrscheinlichsten dichten Werte angesichts der spärlichen Werte zu schätzen. Bei einer beispielhaften Verdichtungstechnik unter Verwendung eines DNN kann ein spärliches Projektionsbild in ein DNN eingegeben werden, um ein entsprechendes dichtes Projektionsbild vorherzusagen. Trainingsdaten für ein solches DNN können auf verschiedene Weise erzeugt und zum Trainieren des DNN verwendet werden, um eine dichte Darstellung der 3D-Oberflächenstruktur anhand einer spärlichen Darstellung vorherzusagen. Zu den Beispieltechniken für die Erzeugung von Trainingsdaten gehören 1) das Rendem von Frames virtueller

Sensordaten, Segmentierungsmasken und Tiefenkarten; 2) die parametrische mathematische Modellierung einer 3D-Straßenoberfläche; 3) das Sammeln und Annotieren realer Sensordaten von einem einzelnen LiDAR-Sensor; und/oder 4) das Sammeln und Annotieren realer Sensordaten, die von mehreren LiDAR-Sensoren gesammelt wurden.

[0006] Von daher können die hierin beschriebenen Techniken zur Beobachtung und Rekonstruktion einer 3D-Oberfläche, z. B. einer 3D-Straßenoberfläche, verwendet werden, und eine Darstellung der 3D-Oberflächenstruktur (und/oder entsprechende Konfidenzwerte) kann einem autonomen Fahrzeugfahrstapel zur Verfügung gestellt werden, um eine sichere und komfortable Planung und Steuerung des autonomen Fahrzeugs zu ermöglichen. Zum Beispiel kann ein autonomes Fahrzeug das Aufhängungssystem des Fahrzeugs an die aktuelle Straßenoberfläche anpassen (z. B. durch Kompensation von Unebenheiten auf der Straße). In einem anderen Beispiel kann ein autonomes Fahrzeug so navigieren, dass es Unebenheiten (z. B. Vertiefungen, Löcher) auf der Straße ausweicht. In einem weiteren Beispiel kann ein autonomes Fahrzeug eine frühzeitige Beschleunigung oder Abbremsung basierend auf einem sich nähernden Neigung auf der Straße vornehmen. Jede dieser Funktionen kann dazu dienen, die Sicherheit zu erhöhen, die Langlebigkeit des Fahrzeugs zu verbessern, die Energieeffizienz zu steigern und/oder ein angenehmes Fahrgefühl zu vermitteln.

[0007] Diese Zusammenfassung wird bereitgestellt, um eine Auswahl von Konzepten in vereinfachter Form vorzustellen, die weiter unten in der ausführlichen Beschreibung beschrieben werden. Diese Zusammenfassung dient nicht dazu, Schlüsselmerkmale oder wesentliche Merkmale zu identifizieren.

Figurenliste

[0008] Die vorliegenden Systeme und Verfahren zur 3D-Oberflächenabschätzung werden im Folgenden unter Bezugnahme auf die beigefügten Zeichnungsfiguren detailliert beschrieben, wobei:

Fig. 1 ist ein Datenflussdiagramm, das eine beispielhafte 3D-Oberflächenrekonstruktionspipeline gemäß einigen Ausführungsformen der vorliegenden Offenbarung darstellt;

Fig. 2 ist ein Diagramm, das einen beispielhaften 3D-Strukturschätzer gemäß einigen Ausführungsformen der vorliegenden Offenbarung darstellt;

Fig. 3 ist ein Diagramm, das einen beispielhaften Erkennungsverdichter gemäß einigen Ausführungsformen der vorliegenden Offenbarung darstellt;

Fig. 4 ist ein Diagramm, das einen beispielhaften ungerichteten Graphen darstellt, der die Beziehung zwischen spärlichen und dichten Höhenkarten modelliert, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 5 ist ein Datenflussdiagramm, das einen beispielhaften Deep-Learning-Modell-Oberflächenschätzer gemäß einigen Ausführungsformen der vorliegenden Offenbarung darstellt;

Fig. 6 ist ein Datenflussdiagramm, das einen beispielhaften Deep-Learning-Modell-Oberflächenschätzer darstellt, der ein Deep-Learning-Modell bzw. mehrere Deep-Learning-Modelle mit mehreren Eingangsköpfen gemäß einigen Ausführungsformen der vorliegenden Offenbarung enthält;

Fig. 7 ist ein Flussdiagramm, das ein Verfahren zum Erzeugen einer Darstellung einer dreidimensionalen (3D) Oberflächenstruktur während einer Erfassungssitzung darstellt, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 8 ist ein Flussdiagramm, das ein Verfahren zum Erzeugen einer verdichteten Darstellung einer 3D-Oberflächenstruktur darstellt, die mindestens auf einem Markov-Zufallsfeld basiert, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 9 ist ein Flussdiagramm, das ein Verfahren zur Steuerung eines Fahrzeugs darstellt, das mindestens teilweise auf einer 3D-Straßenoberflächenstruktur basiert, die unter Verwendung eines oder mehrerer neuronaler Netze geschätzt wird, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 10 ist ein Datenflussdiagramm, das eine beispielhafte Trainingsdaten-Erzeugungspipeline darstellt, die eine simulierte Umgebung verwendet, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 11 ist eine Darstellung eines beispielhaften parametrischen mathematischen Modells einer gewünschten Oberfläche, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 12 ist ein Datenflussdiagramm, das eine beispielhafte Ground-Truth-Erzeugungspipeline darstellt, die gesammelte reale Daten verwendet, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 13A ist eine Darstellung von LiDAR-Daten aus einem beispielhaften LiDAR-Scan, und **Fig. 13B** ist eine Darstellung von LiDAR-Daten, die aus mehreren LiDAR-Scans akkumuliert wurden, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 14 ist ein Flussdiagramm, das ein Verfahren zum Trainieren eines oder mehrerer neuronaler Netze (NNs) darstellt, um eine verdichtete Darstellung der 3D-Oberflächenstruktur unter Verwendung simulierter Bilddaten zu erzeugen, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 15 ist ein Flussdiagramm, das ein Verfahren zum Erzeugen unvollständiger und Ground-Truth-Darstellungen einer synthetischen 3D-Straßenoberfläche für einen Trainingsdatensatz gemäß einigen Ausführungsformen der vorliegenden Offenbarung darstellt;

Fig. 16 ist ein Flussdiagramm, das ein Verfahren zum Trainieren eines oder mehrerer neuronaler Netze (NNs) darstellt, um eine verdichtete Darstellung der 3D-Oberflächenstruktur unter Verwendung von Bilddaten und LiDAR-Daten zu erzeugen, die während einer Erfassungssitzung erfasst werden, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 17A stellt ein Beispiel für ein autonomes Fahrzeug gemäß einigen Ausführungsformen der vorliegenden Offenbarung dar;

Fig. 17B ist ein Beispiel für Kamerapositionen und Sichtfelder für das autonome Fahrzeug der **Fig. 17A**, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 17C ist ein Blockdiagramm einer beispielhaften Systemarchitektur für das beispielhafte autonome Fahrzeug der **Fig. 17A**, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 17D ist ein Systemdiagramm für die Kommunikation zwischen Cloudbasierten Server(n) und dem beispielhaften autonomen Fahrzeug der **Fig. 17A**, gemäß einigen Ausführungsformen der vorliegenden Offenbarung;

Fig. 18 ist ein Blockdiagramm einer beispielhaften Rechenvorrichtung, die zur Verwendung beim Implementieren einiger Ausführungsformen der vorliegenden Offenbarung geeignet ist; und

Fig. 19 ist ein Blockdiagramm eines beispielhaften Datenzentrums, das sich zur Verwendung beim Implementieren einiger Ausführungsformen der vorliegenden Offenbarung geeignet ist.

Detaillierte Beschreibung

[0009] Es werden Systeme und Verfahren zur dreidimensionalen (3D) Oberflächenabschätzung offenbart. Zum Beispiel beschreibt die vorliegende Offenbarung Systeme und Verfahren zur Rekonstruktion einer 3D-Oberflächenstruktur einer Straße oder einer anderen Komponente einer Umgebung zur Verwendung durch autonome Fahrzeuge, halbautonome Fahrzeuge, Roboter und/oder andere Objekttypen. Obwohl die vorliegende Offenbarung in Bezug auf ein Beispiel für ein autonomes Fahrzeug 1700 (hier alternativ als „Fahrzeug 1700“ oder „Ego-Fahrzeug 1700“ bezeichnet, wovon ein Beispiel in Bezug auf die **Fig. 17A-17D** beschrieben ist), ist dies nicht als Einschränkung zu verstehen. Beispielsweise können die hier beschriebenen Systeme und Verfahren ohne Einschränkung von nicht-autonomen Fahrzeugen, halbautonomen Fahrzeugen (z. B. in einem oder mehreren adaptiven Fahrerassistenzsystemen (ADAS)), gelenkten und ungelenkten Robotern oder Roboterplattformen, Lagerfahrzeugen, Geländefahrzeugen, Fahrzeugen, die mit einem oder mehreren Anhängern gekoppelt sind, Flugschiffen, Booten, Shuttles, Notarzteinsatzfahrzeugen, Motorrädern, elektrischen oder motorisierten Fahrrädern, Flugzeugen, Baufahrzeugen, Unterwasserfahrzeugen, Drohnen und/oder anderen Fahrzeugtypen verwendet werden. Obwohl einige Ausführungsformen in Bezug auf die 3D-Oberflächenstrukturschätzung für das autonome Fahren beschrieben werden, soll dies nicht einschränkend sein, und die hier beschriebenen Systeme und Verfahren können in den Bereichen erweiterter Realität, virtueller Realität, gemischter Realität, Robotik, Sicherheit und Überwachung, autonome oder halbautonome Maschinenanwendungen und/oder in jedem anderen Technologiebereich eingesetzt werden, in dem die 3D-Oberflächenstrukturschätzung verwendet werden kann.

[0010] Auf einer hohen Ebene kann eine 3D-Oberflächenstruktur wie die 3D-Oberflächenstruktur einer Straße (3D-Straßenoberfläche) beobachtet und geschätzt werden, um eine 3D-Punktwolke oder eine andere Darstellung der 3D-Oberflächenstruktur zu erzeugen. Da die Darstellung spärlich sein kann, können eine oder mehrere Verdichtungstechniken angewendet werden, um eine dichte Darstellung der 3D-Oberflächenstruktur zu erzeugen, die einem autonomen Fahrzeugfahrstapel zur Verfügung gestellt werden kann, um eine sichere und komfortable Planung und Steuerung des autonomen Fahrzeugs zu ermöglichen.

[0011] In einem beispielhaften Verfahren können eine oder mehrere Kameras an einem Fahrzeug oder einem anderen Objekt befestigt oder anderweitig angeordnet werden und dazu verwendet werden, Bilder einer 3D-Umgebung aufzunehmen, während das Fahrzeug oder Objekt (z. B. entlang einer Straße) durch die 3D-Umgebung navigiert, und die Bilder können dazu verwendet werden, die 3D-Oberflächenstruktur der Straße zu schätzen. Jedes geeignete Verfahren zur Schätzung der 3D-Struktur kann angewendet werden. So kann z. B. Structure from Motion (SFM) zur Schätzung der 3D-Struktur aus Bildsequenzen durchgeführt werden und/oder Stereovision zur Schätzung der 3D-Struktur aus Bildern

angewendet werden, die von mehreren Kameras und/oder aus mehreren Perspektiven aufgenommen wurden. Im Allgemeinen kann die 3D-Strukturschätzung eine Darstellung der erkannten Punkte in der 3D-Umgebung erzeugen, wie z. B. eine 3D-Punktwolke. In einigen Ausführungsformen werden Ausreißer mit Hilfe einer statistischen oder Clustertechnik entfernt. In einigen Fällen kann die Freiraumschätzung auf die aufgenommenen Bilder angewendet werden, um die Straße oder einen anderen befahrbaren Bereich zu erkennen, und eine Segmentierungsmaske oder eine andere Darstellung der erkannten Straße oder des befahrbaren Bereichs kann verwendet werden, um 3D-Punkte auf der Straßenoberfläche auszuwählen (z. B. können Punkte außerhalb der Straßenoberfläche herausgefiltert werden). Das Ergebnis kann eine Darstellung der 3D-Oberflächenstruktur der Straße sein, wie z. B. eine 3D-Punktwolke.

[0012] Aufgrund von Beschränkungen der Genauigkeit und der Rechenleistung der 3D-Strukturschätzverfahren kann die Darstellung der 3D-Oberflächenstruktur der Straße spärlich sein. Daher kann in einigen Ausführungsformen die Darstellung der 3D-Oberflächenstruktur verdichtet werden, wobei als nicht einschränkende Beispiele ein Markov-Zufallsfeld und/oder ein tiefes neuronales Netz (DNN) verwendet wird. In einigen Fällen kann die 3D-Oberflächenstruktur als 2D-Höhenkarte dargestellt werden. Beispielsweise kann eine spärliche 3D-Punktwolke projiziert werden, um ein Projektionsbild (z. B. ein Top-Down-Projektionsbild) zu bilden, das spärliche Erfassungen darstellt, und das Projektionsbild (z. B. eine 2D-Höhenkarte) kann verdichtet werden, um fehlende Werte zu ergänzen.

[0013] Bei einer beispielhaften Verdichtungstechnik, die ein Markov-Zufallsfeld verwendet, können die spärlichen und dichten Projektionsbilder mit einem ungerichteten Graphen modelliert werden, und es kann eine Maximum-a-Posterior (MAP)-Inferenz verwendet werden, um die wahrscheinlichsten dichten Werte anhand der spärlichen Werte zu schätzen. Zum Beispiel kann jedes Pixel im dichten Projektionsbild (z. B. die dichte 2D-Höhenkarte) mit einem entsprechenden Knoten modelliert werden, der Kanten aufweist, die mit benachbarten Knoten verbunden sind (z. B. eine Kante für jedes benachbarte Pixel). Jedes Pixel des spärlichen Projektionsbildes (z. B. die spärliche 2D-Höhenkarte) kann als eine verrauschte Beobachtung des dichten Projektionsbildes betrachtet und als ein Knoten mit einer Kante modelliert werden, die eine Verbindung zu einem entsprechenden Knoten (Pixel) aus dem dichten Projektionsbild herstellt. Nehmen wir zum Beispiel an, dass der Graph Knoten hat, die zwei Schichten eines Gitters bilden, wobei die untere Schicht der Ground Truth entspricht (das dichte Projektionsbild) und die obere Schicht einer verrauschten Beobachtung (das spärliche Projektionsbild) entspricht. Unter der Annahme, dass jeder Knoten im Graphen einer Zufallsvariablen entspricht, kann das Markov-Zufallsfeld für den Graphen eine gemeinsame Wahrscheinlichkeitsverteilung der den Knoten im Graphen entsprechenden Zufallsvariablen modellieren oder anderweitig darstellen. Bei Kenntnis der gemeinsamen Wahrscheinlichkeitsverteilung und eines Satzes beobachteter Werte (aus dem spärlichen Projektionsbild) können Werte für das dichte Projektionsbild (z. B. eine Höhenschätzung für jedes Pixel der dichten 2D-Höhenkarte) unter Verwendung eines beliebigen bekannten MAP-Inferenzalgorithmus geschätzt werden, wie z. B. Iterative Conditional Mode, Gaussian Belief Propagation oder andere. So kann ein Markov-Zufallsfeld verwendet werden, um die Darstellung der 3D-Oberflächenstruktur zu verdichten.

[0014] In einigen Ausführungsformen kann ein tiefes neuronales Netz (DNN) verwendet werden, um Werte für eine dichte Darstellung der 3D-Oberflächenstruktur vorherzusagen. Beispielsweise kann eine spärliche 3D-Punktwolke projiziert werden, um ein spärliches Projektionsbild (z. B. ein Top-Down-Projektionsbild) zu bilden, das spärliche Erfassungen darstellt, und das spärliche Projektionsbild kann in ein DNN, wie z. B. ein neuronales Faltungsnetz (CNN) eingespeist werden, um ein dichtes Projektionsbild vorherzusagen. In einigen Ausführungsformen kann das DNN einen gemeinsamen Stamm (oder einen Strom von Schichten) enthalten, der mit einem oder mehreren Köpfen (oder mindestens teilweise diskreten Strömen von Schichten) verbunden ist, die verschiedene Ausgaben vorhersagen. Zum Beispiel kann ein Regressionskopf eine bestimmte Art von Informationen über die 3D-Oberflächenstruktur regressieren, wie z. B. einen Höhenwert für jedes Pixel. In einigen Ausführungsformen kann ein Konfidenzkopf eine Konfidenzkarte mit Werten vorhersagen, die die Konfidenz eines entsprechenden vom Regressionskopf vorhergesagten regressierten Wertes repräsentiert. Von daher und wie im Folgenden näher erläutert, kann das DNN so trainiert werden, dass er eine dichte Darstellung der 3D-Oberflächenstruktur (z. B. eine dichte 2D-Höhenkarte) und/oder eine entsprechende Konfidenzkarte vorhersagt.

[0015] In einigen Ausführungsformen kann ein spärliches Projektionsbild normalisiert werden, bevor es in das DNN eingegeben wird. Zum Beispiel kann eine 2D-Höhenkarte Höhenwerte speichern, die einen systematischen Fehler enthalten, der der Höhe der Kamera entspricht, die die Bilder aufgenommen hat, aus denen die 2D-Höhenkarte abgeleitet wurde. So kann in einigen Ausführungsformen die mittlere Höhe der Höhenwerte in der 2D-Höhenkarte berechnet und von allen Höhenwerten subtrahiert werden, um den systematischen Fehler zu entfernen, was das Lernen des DNN erleichtern kann. In Ausführungsformen, in denen ein systematischer Fehler

aus der DNN-Eingabe entfernt wird, kann der systematische Fehler wieder in eine vorhergesagte Ausgabe des DNN (z. B. zu den vom Regressionskopf vorhergesagten Werten) eingeführt (z. B. hinzugefügt) werden.

[0016] In einigen Ausführungsformen kann das DNN mehrere Eingabeköpfe (oder mindestens teilweise diskrete Ströme von Schichten) für separate Eingaben enthalten. Zum Beispiel kann das DNN einen ersten Eingangskopf enthalten, der ein spärliches Projektionsbild akzeptiert, und einen zweiten Eingangskopf, der ein RGB-Bild akzeptiert, wie z. B. ein perspektivisches Bild, aus dem das spärliche Projektionsbild erzeugt wurde. Auf diese Weise kann das DNN aus zwei verschiedenen Ansichten des zugrunde liegenden dichten Straßenprofils lernen (z. B. von oben nach unten und perspektivisch, im 3D-Punktwolkenraum und im 2D-Bildraum usw.). In einem solchen Beispiel können die mehreren Eingabeköpfe mit einem gemeinsamen Stamm verbunden sein, der die mehreren Eingabeköpfe zusammenführt. So kann das DNN für multimodales Lernen verwendet werden, indem Informationen aus verschiedenen Quellen für eine bessere Vorhersage zusammengeführt werden.

[0017] In einigen Ausführungsformen kann das DNN eine oder mehrere rekurrente Schichten (z. B. Gated Recurrent Units, Long Short Term Memory) enthalten, um zeitliche Informationen zu nutzen. Die Einbeziehung einer oder mehrerer rekurrenter Schichten kann es dem DNN ermöglichen, Informationen aus früheren Zeitabschnitten zu nutzen, was zu besseren Vorhersagen und stabileren Verdichtungsergebnissen im Zeitverlauf führt.

[0018] Trainingsdaten für das DNN können auf verschiedene Weise erzeugt und zum Trainieren des DNN verwendet werden, um eine dichte Darstellung der 3D-Oberflächenstruktur bei einer spärlichen Darstellung vorherzusagen. Im Allgemeinen können reale Daten und/oder virtuelle Daten gesammelt und verwendet werden, um Trainingsdaten abzuleiten (z. B. spärliche Eingabedaten und/oder Ground-Truth-Darstellungen der 3D-Oberflächenstruktur). Die Art der Trainingsdaten kann von der Implementierung des DNN abhängen. Zu den Eingabetrainingsdaten können beispielsweise spärliche Darstellungen der 3D-Oberflächenstruktur (z. B. spärliche Höhenkarten) und/oder Bilddaten aus einer anderen Perspektive (z. B. Bilder einer perspektivischen Ansicht) gehören. Ground-Truth-Trainingsdaten können dichte Darstellungen der 3D-Oberflächenstruktur (z. B. dichte Höhenkarten) und/oder Segmentierungsmasken umfassen (die z. B. eine gewünschte Oberfläche wie eine Straße oder einen anderen befahrbaren Raum identifizieren).

[0019] Zu den Beispieltechniken für die Erzeugung von Trainingsdaten gehören 1) das Rendern von Frames virtueller Sensordaten, Segmentierungsmasken und Tiefenkarten; 2) die parametrische mathematische Modellierung einer 3D-Straßenoberfläche; 3) das Sammeln und Annotieren echter Sensordaten von einem einzelnen LiDAR-Sensor; und/oder 4) das Sammeln und Annotieren echter Sensordaten, die von mehreren LiDAR-Sensoren gesammelt wurden.

[0020] In einem beispielhaften Verfahren zum Erzeugen von Trainingsdaten kann eine Simulation durchgeführt werden, um Frames virtueller Sensordaten (z. B. Bilder) zu rendern, die realistische Fahrszenarien darstellen, und um entsprechende Segmentierungsmasken (z. B. Ground-Truth-Segmentierungsmasken, die eine gewünschte Oberfläche wie eine Straße oder einen anderen befahrbaren Raum identifizieren) und Tiefenkarten zu erzeugen. Für jedes beliebige gerenderte Bild kann eine 3D-Oberflächenstruktur (z. B. eine 3D-Straßenoberfläche) aus dem Bild geschätzt werden, wie hierin beschrieben, und die sich ergebenden spärlichen Werte können projiziert werden, um ein spärliches Projektionsbild (z. B. eine 2D-Höhenkarte) zu bilden, das als Eingabetrainingsdaten verwendet werden kann.

[0021] Um ein entsprechendes dichtes Ground-Truth-Projektionsbild zu erzeugen, kann für jedes gegebene Bild, das aus der Perspektive eines virtuellen Sensors gerendert wird, eine 3D-Punktwolke oder eine andere Darstellung der 3D-Struktur erzeugt werden, indem Entfernungswerte aus der entsprechenden Tiefenkarte umgekehrt in die 3D-Umgebung unter Verwendung der Position und Ausrichtung des virtuellen Sensors projiziert werden. Die Segmentierungsmaske kann verwendet werden, um 3D-Punkte auf der Straßenoberfläche auszuwählen (z. B. können Punkte außerhalb der Straßenoberfläche herausgefiltert werden). Zusätzlich oder alternativ kann die Segmentierungsmaske verwendet werden, um Punkte aus der Tiefenkarte auszuwählen, die sich auf der Straßenoberfläche befinden, und die ausgewählten Punkte können umgekehrt in die 3D-Umgebung projiziert werden, um die 3D-Punkte auf der Straßenoberfläche zu erzeugen. In einigen Fällen kann die resultierende Darstellung der 3D-Straßenoberfläche immer noch spärlich sein. Daher können in einigen Ausführungsformen fehlende Werte mit einem Triangulationsalgorithmus interpoliert werden. Beispielsweise kann die Delaunay-Triangulierung in 2D (z. B. durch Projektion der 3D-Punkte zur Bildung eines Projektionsbildes und Durchführung der Delaunay-Triangulierung im Projektionsbild) oder in 3D (durch Berechnung eines Oberflächengitters aus Dreiecken, die die 3D-Punktewolke umgeben) durchgeführt werden, und aus den Dreiecken können Punkte abgetastet werden,

um eine gewünschte Anzahl von Punkten für ein dichtes Ground-Truth-Projektionsbild zu erzeugen. Beispielsweise kann eine 2D-Ground-Truth-Höhenkarte aus Dreiecken abgetastet werden, die durch die Durchführung einer 2D-Delaunay-Triangulierung in einer projizierten Höhenkarte erzeugt wurden, oder durch Projizieren von 3D-Punkten, die aus einem Oberflächengitter abgetastet wurden, das durch die Durchführung einer 3D-Delaunay-Triangulierung erzeugt wurde. So kann das dichte Projektionsbild und/oder die Segmentierungsmaske als Ground Truth verwendet, mit dem eingegebenen spärlichen Projektionsbild gepaart und in einen Trainingsdatensatz aufgenommen werden.

[0022] In einer anderen beispielhaften Technik zum Erzeugen von Trainingsdaten können synthetische Trainingsdaten durch parametrische mathematische Modellierung einer 3D-Straßenoberfläche erzeugt werden. In einem Ausführungsbeispiel kann eine synthetische 3D-Straßenoberfläche erzeugt werden, indem zunächst longitudinale Werte abgetastet werden (z. B. von 0 bis 300 m) und dann die laterale Werte als Polynom zweiter Ordnung der longitudinalen Werte berechnet werden, wobei Werte für Polynomkonstanten verwendet werden, die abgetastet werden, um Änderungen der Straßenrichtung zu simulieren (z. B. Linkskurve, Rechtskurve usw.). Die Höhe der synthetischen 3D-Straßenoberfläche kann als Linearkombination von Fourier-Basen berechnet werden, wobei unterschiedliche Abtastwerte für die Anzahl der Basen, die Gewichtung für eine bestimmte Basis und die Frequenz für eine bestimmte Basis verwendet werden, um Änderungen der Oberflächenhöhe zu simulieren. Diese Schritte erzeugen eine longitudinale 3D-Kurve, die zu einer 3D-Oberfläche erweitert werden kann, indem eine laterale 3D-Kurve durch jeden Punkt auf der longitudinalen 3D-Kurve gezogen wird, wobei Abtastwerte für den Winkel zwischen der lateralen 3D-Kurve und der Bodenebene verwendet werden, um Änderungen der lateralen Oberflächenneigung zu simulieren. Jede laterale 3D-Kurve kann abgetastet werden, um eine dichte 3D-Punktwolke zu erzeugen, die projiziert werden kann, um ein synthetisches Ground-Truth-Projektionsbild zu bilden (z. B. eine 2D-Ground-Truth-Höhenkarte).

[0023] Um ein entsprechendes spärliches Projektionsbild für die Eingabetrainingsdaten zu erzeugen, kann ein bekanntes Muster auf das Ground-Truth-Projektionsbild angewendet werden, um eine Teilmenge von Pixelwerten auszulöschen (z. B. indem diese Pixelwerte auf null gesetzt werden), um unbeobachtete Werte zu simulieren. Beispielsweise können Frames von realen Daten gesammelt werden, eine 3D-Oberflächenstruktur (z. B. einer 3D-Straßenoberfläche) kann aus jedem Frame geschätzt werden (wie hierin beschrieben), die geschätzte 3D-Struktur (z. B. eine 3D-Punktwolke) kann projiziert werden, um ein Projektionsbild zu bilden (z. B. eine spärliche 2D-Höhenkarte), und eine entsprechende binäre Karte, die darstellt, welche Pixel des Projektionsbildes vorhanden oder beobachtet sind, kann erzeugt werden. Es können mehrere binäre Karten aus realen Daten erzeugt werden, und eine der binären Karten kann zufällig ausgewählt und mit einem Ground-Truth-Projektionsbild multipliziert werden, um ein entsprechendes synthetisches spärliches Projektionsbild zu erzeugen. Auf diese Weise kann für jedes Ground-Truth-Projektionsbild ein spärliches Projektionsbild erzeugt werden, und die Paare aus synthetischen spärlichen und Ground-Truth-Projektionsbildern können in einen Trainingsdatensatz aufgenommen werden.

[0024] Zusätzlich oder alternativ können die Trainingsdaten aus realen Daten erzeugt werden. Beispielsweise können ein oder mehrere Fahrzeuge Sensordaten von einem ausgestatteten Sensor, wie einer oder mehreren Kameras und LiDAR-Sensoren, sammeln, während sie durch eine reale (z. B. physikalische) Umgebung navigieren. Um Ground-Truth-Trainingsdaten zu erzeugen, können die gesammelten LiDAR-Daten (z. B. LiDAR-Punktwolken) geglättet, Ausreißer entfernt und die LiDAR-Daten zeitlich und/oder räumlich mit den entsprechenden Bildern der Bilddaten ausgerichtet werden. In einigen Ausführungsformen können zur Verdichtung der gesammelten LiDAR-Daten fehlende Werte mit Hilfe der Delaunay-Triangulierung interpoliert und/oder LiDAR-Daten, die von mehreren LiDAR-Sensoren ausgelöst und/oder in derselben Zeitscheibe erfasst wurden, akkumuliert werden, um die gesammelten Daten zu verdichten. Die LiDAR-Daten können gekennzeichnet werden, um 3D-Punkte auf einer interessierenden Oberfläche (z. B. einer 3D-Straßenoberfläche) zu identifizieren, und eine Darstellung der identifizierten 3D-Punkte (z. B. eine 3D-Punktwolke, ein Projektionsbild) kann als Ground-Truth-Trainingsdaten bezeichnet werden. In einigen Ausführungsformen kann ein entsprechender Frame von Bilddaten klassifiziert werden, um eine Ground-Truth-Segmentierungsmaske zu erzeugen, die die gewünschte Oberfläche identifiziert.

[0025] Um entsprechende Eingabetrainingsdaten zu erzeugen, kann eine 3D-Oberflächenstruktur (z. B. eine 3D-Straßenoberfläche) aus einem Frame von Bilddaten (wie hierin beschrieben) geschätzt werden, und eine Darstellung der geschätzten 3D-Struktur (z. B. eine spärliche 3D-Punktewolke, ein spärliches Projektionsbild) kann als Eingabetrainingsdaten bezeichnet werden. Von daher können ein entsprechendes spärliches Projektionsbild, ein Bild aus Bilddaten, ein dichtes Projektionsbild und/oder eine Segmentierungsmaske gruppiert und in einen Trainingsdatensatz aufgenommen werden.

[0026] Während des Trainings kann jede geeignete Verlustfunktion verwendet werden, um die vorhergesagte(n) Ausgabe(n) mit der Ground Truth zu vergleichen und das DNN zu aktualisieren. In einem Ausführungsbeispiel, in dem das DNN einen Regressionskopf enthält, der eine Höhenkarte vorhersagt, kann eine Verlustfunktion die vorhergesagten und die Ground-Truth-Höhenkarten vergleichen und mit einer Ground-Truth-Segmentierungsmaske multiplizieren, die die zu verdichtende Oberfläche angibt, wodurch Aktualisierungen des DNN basierend auf Vorhersagen, die außerhalb des zu verdichtenden Bereichs auftreten, effektiv aufgehoben werden. In diesem Beispiel kann das DNN lernen, Höhenkarten unter Verwendung von Ground-Truth-Höhenkarten und Segmentierungsmasken vorherzusagen. In einer anderen Ausführungsform, in der das DNN einen Regressionskopf, der eine Höhenkarte vorhersagt, und einen Konfidenzkopf enthält, der eine der Höhenkarte entsprechende Konfidenzkarte vorhersagt, kann eine Verlustfunktion die vorhergesagten Höhen mit denen der Ground Truth vergleichen und basierend auf den vorhergesagten Konfidenzwerte kompensieren. In diesem Beispiel kann das DNN lernen, sowohl die Höhen- als auch die Konfidenzkarten aus den Ground-Truth-Höhenkarten vorherzusagen. So kann das DNN lernen, wie es eine Verdichtung durchführt, indem es eine Zuordnung zwischen spärlichen und dichten Darstellungen der 3D-Struktur lernt.

[0027] Von daher können die hier beschriebenen Techniken zur Beobachtung und Rekonstruktion einer 3D-Oberfläche, z. B. einer 3D-Straßenoberfläche, verwendet werden, und eine Darstellung der 3D-Oberflächenstruktur (und/oder entsprechende Konfidenzwerte) kann einem autonomen Fahrzeugfahrstapel zur Verfügung gestellt werden, um eine sichere und komfortable Planung und Steuerung des autonomen Fahrzeugs zu ermöglichen. Im Allgemeinen können die hier beschriebenen Techniken eine genauere Darstellung von Straßenoberflächen erzeugen als frühere Rekonstruktionstechniken. Darüber hinaus kann mit den hier vorgestellten Techniken eine Darstellung von Straßenoberflächen mit ausreichender Genauigkeit und Reichweite für bestimmte Anwendungen des autonomen Fahrens erzeugt werden, im Gegensatz zu früheren Rekonstruktionstechniken. So kann die Darstellung von Straßenoberflächen, die mit den vorliegenden Techniken erzeugt wird, eine verbesserte Navigation, Sicherheit und Komfort beim autonomen Fahren ermöglichen. So kann ein autonomes Fahrzeug beispielsweise besser in der Lage sein, das Aufhängungssystem des Fahrzeugs an die aktuelle Straßenoberfläche anzupassen (z. B. durch Kompensation von Unebenheiten auf der Straße), das Fahrzeug so zu steuern, dass es Erhebungen (z. B. Vertiefungen, Löcher) auf der Straße ausweicht, und/oder eine frühzeitige Beschleunigung oder Abbremsung basierend auf einer sich nähernden Oberflächenneigung auf der Straße vorzunehmen. Jede dieser Funktionen kann dazu dienen, die Sicherheit zu erhöhen, die Langlebigkeit des Fahrzeugs zu verbessern, die Energieeffizienz zu steigern und/oder ein angenehmes Fahrgefühl zu vermitteln.

Beispielhaft 3D-Oberflächenrekonstruktionspipeline

[0028] Mit Bezug auf **Fig. 1** ist **Fig. 1** ein Datenflussdiagramm, das eine beispielhafte 3D-Oberflächenrekonstruktionspipeline 100 für ein 3D-Oberflächenrekonstruktionssystem gemäß einigen Ausführungsformen der vorliegenden Offenbarung zeigt. Es sollte verstanden werden, dass diese und andere hier beschriebene Anordnungen nur als Beispiele dargestellt werden. Andere Anordnungen und Elemente (z. B. Maschinen, Schnittstellen, Funktionen, Reihenfolgen, Gruppierungen von Funktionen usw.) können zusätzlich zu oder anstelle der gezeigten verwendet werden, und einige Elemente können ganz weggelassen werden. Außerdem sind viele der hier beschriebenen Elemente funktionale Einheiten, die als einzelne oder verteilte Komponenten oder in Verbindung mit anderen Komponenten und in jeder geeigneten Kombination und an jedem geeigneten Ort implementiert werden können. Verschiedene hier beschriebene Funktionen, die von Einheiten ausgeführt werden, können von Hardware, Firmware und/oder Software ausgeführt werden. Beispielsweise können verschiedene Funktionen von einem Prozessor ausgeführt werden, der im Speicher gespeicherte Anweisungen ausführt.

[0029] Auf einer hohen Ebene kann die Pipeline 100 eine Darstellung einer beobachteten 3D-Oberflächenstruktur, wie die einer 3D-Straßenoberfläche oder eines anderen Umweltteils, basierend auf Bilddaten 102 einer dreidimensionalen (3D) Umgebung schätzen und erzeugen. Die Bilddaten 102 können von einer oder mehreren Kameras 101 eines Ego-Objekts oder Ego-Akteurs (z. B. dem autonomen Fahrzeug 1700 der **Fig. 17A-17D**, auch als Fahrzeug 1700 bezeichnet) erfasst werden, während das Ego-Objekt oder der Ego-Akteur durch die 3D-Umgebung navigiert. Ein 3D-Strukturschätzer 105 kann die Bilddaten 102 verarbeiten, um eine Darstellung einer interessierenden 3D-Oberflächenstruktur (z. B. spärliche Erkennungsdaten 110) zu erzeugen, die eine 3D-Punktwolke umfassen kann. Da die geschätzte 3D-Oberflächenstruktur spärlich sein kann, kann ein Erkennungsverdichter 115 die spärlichen Erkennungsdaten 110 verdichten, um eine dichtere Darstellung der 3D-Oberflächenstruktur (z. B. dichte Erkennungsdaten 120) zu erzeugen, die eine zweidimensionale (2D) Top-down-Höhenkarte und/oder eine 3D-Punktwolke umfassen kann. Die dichten Erfassungsdaten 120 können die beobachtete 3D-Oberflächenstruktur darstellen, z. B. eine 3D-Straßenoberfläche oder einen anderen Teil der Umge-

bung. Von daher können die dichten Erkennungsdaten 120 oder eine andere Darstellung der beobachteten 3D-Oberflächenstruktur den Steuerkomponente(n) des Ego-Objekts oder Ego-Akteurs (z. B. Software-Stack 122 und/oder Komponenten des autonomen Fahrzeugs 1700 der **Fig. 17A-17D**, wie z. B. Controller 1736, ADAS-System 1738 und/oder SOC(s) 1704) bereitgestellt und verwendet werden, um das Ego-Objekt oder den Ego-Aktor bei der Durchführung einer oder mehrerer Operationen innerhalb der 3D-Umgebung zu unterstützen, wie z. B. der Wegplanung, der Vermeidung von Hindernissen oder Höckern, der Anpassung eines Aufhängungssystems des Ego-Objekts oder des Ego-Aktors an die aktuelle Straßenoberfläche, der Anwendung einer frühzeitigen Beschleunigung oder Abbremsung basierend auf einer sich nähernden Oberflächenneigung, der Kartierung und/oder anderen.

[0030] Im Allgemeinen kann die 3D-Oberflächenrekonstruktion unter Verwendung von Bilddaten 102 von einer beliebigen Anzahl und einem beliebigen Typ von Kameras (z. B. der Kamera(s) 101) durchgeführt werden, wie sie im Folgenden in Bezug auf das autonome Fahrzeug 1700 der **Fig. 17A-17D** beschrieben werden. Beispielsweise können die Kamera(s) 101 eine oder mehrere Kameras eines Ego-Objekts oder Ego-Akteurs umfassen, wie beispielsweise Stereokamera(s) 1768, Weitwinkelkamera(s) 1770 (z. B. Fischaugenkameras), Infrarotkamera(s) 1772, Surround-Kamera(s) 1774 (z. B. 360-Grad-Kameras) und/oder Fern- und/oder Mittelbereichskamera(s) 1798 des autonomen Fahrzeugs 1700 der **Fig. 17A-17D** - und die Kamera(s) 101 können verwendet werden, um die Bilddaten 102 der 3D-Umgebung um das Ego-Objekt oder den Ego-Akteur zu erzeugen. In Ausführungsformen, in denen mehrere Kameras verwendet werden, können die mehreren Kameras einen gemeinsamen Bereich der 3D-Umgebung mit einem überlappenden Teil ihrer jeweiligen Sichtfelder betrachten, so dass die Bilddaten 102 (z. B. Bilder) von verschiedenen Kameras den gemeinsamen Bereich darstellen.

[0031] Der 3D-Strukturschätzer 105 schätzt die 3D-Struktur einer bestimmten Oberfläche (z. B. spärliche Erkennungsdaten 110) aus den Bilddaten 102 unter Verwendung von Structure from Motion (SfM), Stereovision und/oder einer anderen 3D-Oberflächenstrukturschätztechnik. SfM und Stereovision sind Entfernungsmesstechniken, die die 3D-Struktur aus mehreren Bildern schätzen. SfM schätzt die 3D-Struktur aus Bildsequenzen (z. B. von derselben Kamera 101 aufgenommen), während Stereovision die 3D-Struktur aus mehreren Bildern schätzt, die im Wesentlichen zur gleichen Zeit aus verschiedenen Perspektiven (z. B. von verschiedenen Kameras 101) aufgenommen wurden. In einigen Ausführungsformen kann auf die Bilddaten 102 vor der Schätzung der 3D-Struktur eine Bildentkrümmung und/oder Verzerrungskorrektur angewendet werden. Eine Segmentierungsmaske oder andere Klassifizierungsdaten können verwendet werden (z. B. durch Überlagerung der Klassifizierungsdaten mit den Bilddaten 102), um Punkte aus der geschätzten 3D-Struktur auszuwählen, die sich auf einer gewünschten Oberfläche befinden, z. B. der Straßenoberfläche. Von daher kann der 3D-Strukturschätzer 105 eine Darstellung der 3D-Struktur einer gewünschten Oberfläche (z. B. spärliche Erkennungsdaten 110) erzeugen, die eine 3D-Punktwolke (z. B. in 3D-Weltkoordinaten) enthalten kann.

[0032] **Fig. 2** ist ein Diagramm, das eine Beispielimplementierung des 3D-Strukturschätzers 105 gemäß einigen Ausführungsformen der vorliegenden Offenbarung zeigt. In **Fig. 2** umfasst der 3D-Strukturschätzer 105 den Structure-from-Motion-Schätzer 210, den Stereovision-Schätzer 230, den Ausreißerentferner 220 und den Straßenoberflächen-Punktselektor 240.

[0033] Der Structure-from-Motion-Schätzer 210 kann jede bekannte SfM-Technik anwenden, um die 3D-Struktur aus den Bilddaten 102 zu schätzen. Beispielsweise kann der Structure-from-Motion-Schätzer 210 3D-Positionen von Merkmalen, die in den Bilddaten 102 dargestellt sind, aus Merkmalstrajektorien rekonstruieren, die im Laufe der Zeit erfasst wurden. In einigen Ausführungsformen kann der Structure-from-Motion-Schätzer 210 eine direkte Schätzung von 3D-Positionen ohne Zwischenschätzung von Merkmalstrajektorien durchführen. Im Allgemeinen kann jede bekannte SfM-Technik angewendet werden, einschließlich inkrementeller SfM, globaler SfM, Out-of-Core-SfM und/oder anderer. Von daher kann der Structure-from-Motion-Schätzer 210 eine Darstellung der 3D-Struktur der in den Bilddaten 102 dargestellten Merkmale erzeugen.

[0034] Der Stereovision-Schätzer 230 kann die 3D-Struktur schätzen, indem er Stereovision (oder einen Stereo-Algorithmus) auf Bilddaten 102 anwendet, die verschiedene Perspektiven repräsentieren. Beispielsweise kann der Stereovision-Schätzer 230 Bilddaten 102 von mehreren Kameras (z. B. die Kamera(s) 101) in einen gemeinsamen Bildraum oder eine gemeinsame Ebene projizieren und die projizierten Bilddaten unter Verwendung einer geeigneten Metrik vergleichen, um eine Disparitätskarte zu erzeugen, die Unterschiede in der Tiefe darstellen kann (z. B. in Bildkoordinaten, die umgekehrt proportional zur Tiefe sein können). Die Disparitätskarte kann unter Verwendung der bekannten Position und Ausrichtung der mehreren Kameras in eine 3D-Punktwolke projiziert werden. Auf diese Weise kann der Stereovision-Schätzer 230 eine Darstellung der 3D-Struktur der in den Bilddaten 102 dargestellten Merkmale erzeugen.

[0035] In einigen Ausführungsformen kann der Ausreißerentferner 220 die geschätzte 3D-Struktur auswerten und Ausreißer entfernen. In einigen Ausführungsformen, in denen die geschätzte 3D-Struktur die Form einer 3D-Punktwolke annimmt, kann die 3D-Punktwolke beispielsweise projiziert werden, um ein Projektionsbild zu erstellen, wie z. B. ein Top-Down-Projektionsbild, um Punktesäulen (z. B. 0,1 Meter × 0,1 Meter große Balken) zu erzeugen. Anschließend kann für jede Spalte ein geeignetes statistisches Verfahren oder eine Clustering-Technik angewendet werden, um einen repräsentativen Punkt für die Spalte zu ermitteln. Als nicht einschränkendes Beispiel kann ein Median- oder Mittelwert der Punkte in einer Spalte als repräsentativer Punkt für die Spalte ermittelt werden. Am Beispiel des Top-Down-Projektionsbildes kann der Median oder der Mittelwert der Höhe der Punkte in jeder Spalte als die Höhe eines repräsentativen Punktes für die Spalte identifiziert werden. In einigen Ausführungsformen kann eine andere Clustering-Technik angewendet werden, um Punkte aus einer 3D-Punktwolke zu gruppieren und repräsentative Punkte (z. B. Clusterzentren oder Mittelwerte) zu identifizieren. Von daher kann der Ausreißerentferner 220 die geschätzte 3D-Struktur mit den identifizierten Punkten aktualisieren und/oder auf andere Weise Ausreißer erkennen und entfernen.

[0036] Im Allgemeinen kann die geschätzte 3D-Struktur 3D-Punkte von Teilen der 3D-Umgebung und Objekten in der 3D-Umgebung enthalten, die in den Bilddaten 102 dargestellt sind. So kann der Straßenoberflächen-Punktselektor 240 Punkte identifizieren, die zu einer bestimmten interessierenden Oberfläche gehören, wie z. B. einer 3D-Straßenoberfläche oder einem anderen Teil der Umgebung. Beispielsweise kann eine Segmentierungsmaske oder andere Klassifizierungsdaten erzeugt oder auf andere Weise erhalten werden, und der Straßenoberflächen-Punktselektor 240 kann die Segmentierungsmaske oder andere Klassifizierungsdaten verwenden, um die Punkte auszuwählen.

[0037] Insbesondere können in einigen Ausführungsformen Objekterkennung, Freiraumschätzung und/oder Bildsegmentierung angewendet werden (z. B. durch den 3D-Schätzer 105 oder eine andere Komponente), um Regionen (z. B. Pixel) der Bilddaten 102 zu klassifizieren, zu segmentieren und/oder vorherzusagen, die Teil einer gewünschten Klasse sind. Beispielsweise können mehrere Deep-Learning-Modelle (z. B. ein neuronales Faltungsnetzwerk) trainiert werden, um eine oder mehrere Segmentierungsmasken und/oder Konfidenzkarten vorherzusagen, die Pixel repräsentieren, die zu einer befahrbaren Straßenoberfläche oder einem anderen befahrbaren Raum, anderen Teilen der Umgebung (z. B. Bürgersteige, Gebäude), belebten Objekten und/oder anderen Klassen gehören. In einigen Ausführungsformen kann ein einzelnes Bild (z. B. ein RBG-Bild), das von einer einzelnen Kamera aufgenommen wurde, segmentiert und/oder klassifiziert werden. In einigen Fällen kann ein zusammengesetztes Bild (z. B. ein RBG-Bild) durch Zusammenfügen von Bildern erzeugt werden, die von mehreren Kameras aufgenommen wurden, und das zusammengesetzte Bild kann segmentiert und/oder klassifiziert werden. Von daher können eine Segmentierungsmaske oder andere Klassifizierungsdaten, die die Straße oder den befahrbaren Raum (oder eine andere gewünschte Oberfläche) abgrenzen oder darstellen, erhalten und/oder erzeugt werden (z. B. aus den vorhergesagten Masken oder Konfidenzkarten).

[0038] So kann der Straßenoberflächen-Punktselektor 240 die Segmentierungsmaske oder andere Klassifizierungsdaten verwenden, um Punkte aus der geschätzten 3D-Struktur auszuwählen, die zu der durch die Segmentierungsmaske oder andere Klassifizierungsdaten repräsentierten Klasse gehören. Es kann jede geeignete Auswahltechnik angewendet werden. In einigen Ausführungsformen können 3D-Punkte aus der geschätzten 3D-Struktur in die Segmentierungsmaske zurückprojiziert werden (z. B. unter Verwendung des bekannten Standorts und der bekannten Ausrichtung der Kamera 101, die die Bilddaten 102 aufgenommen hat, aus denen die Segmentierungsmaske erzeugt wurde), und projizierte Punkte, die innerhalb der vorhergesagten Region liegen, können ausgewählt werden (und/oder projizierte Punkte, die außerhalb der vorhergesagten Region liegen, können entfernt werden). So kann der Straßenoberflächen-Punktselektor 240 die Punkte der geschätzten 3D-Oberflächenstruktur erzeugen oder anderweitig identifizieren, die zu einer gewünschten Oberfläche, wie der 3D-Straßenoberfläche, gehören. In Ausführungsformen, die eine Ausreißerentfernung durchführen, können der Ausreißerentferner 220 und der Straßenoberflächen-Punktselektor 240 in beliebiger Reihenfolge aufgerufen werden. Die resultierende Darstellung der geschätzten 3D-Oberflächenstruktur (z. B. die spärlichen Erkennungsdaten 110) kann eine beliebige Form annehmen, wie z. B. eine 3D-Punktwolke.

[0039] Obwohl bestimmte Ausführungsformen beschrieben werden, bei denen die 3D-Oberflächenrekonstruktion die von der/den Kamera(s) 101 erfassten Bilddaten 102 verwendet, können in einigen Ausführungsformen zusätzlich oder alternativ auch andere Sensordaten verwendet werden. Als nicht einschränkendes Beispiel können ein oder mehrere LiDAR-Sensoren oder RADAR-Sensoren verwendet werden, um spärliche Erfassungsdaten 110 (z. B. eine LiDAR- oder RADAR-Punktwolke) zu erfassen.

[0040] Auf **Fig. 1** zurückkommend, kann der Erkennungsverdichter 115, da die geschätzte 3D-Oberflächenstruktur spärlich sein kann, die spärlichen Erkennungsdaten 110 verdichten, um eine dichtere Darstellung der 3D-Oberflächenstruktur (z. B. die dichten Erkennungsdaten 120) zu erzeugen. Im Allgemeinen können die spärlichen Erkennungsdaten 110 jede geeignete Form annehmen, wie z. B. eine spärliche 3D-Punktwolke. Die spärlichen Erkennungsdaten 110 können projiziert werden, um ein Projektionsbild zu bilden, z. B. eine zweidimensionale (2D) Top-Down-Höhenkarte $o \in N^{m \times n}$ mit fehlenden Werten. Die Schreibweise $N^{m \times n}$ steht für ein Projektionsbild (z. B. ein Überkopfbild) mit den räumlichen Abmessungen mxn (z. B. in Pixeln) und mit einem gewünschten Bodenabtastabstand, wobei jedes Pixel im Projektionsbild einen Fließkommawert (z. B. einen Höhenwert) speichern kann. Diese spärliche 2D-Höhenkarte kann als eine teilweise verrauschte Beobachtung der 3D-Oberflächenstruktur betrachtet werden. In diesem Beispiel können die dichten Erkennungsdaten 120 die Form einer 2D-Top-Down-Höhenkarte $g \in N^{m \times n}$ annehmen oder auf andere Weise repräsentieren, und der Erkennungsverdichter 115 kann die spärlichen Erkennungsdaten durch Inferenz von g verdichten, wenn o gegeben ist. In einigen Ausführungsformen kann der Erkennungsverdichter 115 diese Inferenz unter Verwendung eines oder mehrerer maschineller Lernmodelle, wie z. B. eines Markov-Zufallsfeldes und/oder eines oder mehrerer Deep-Learning-Modelle (z. B. eines oder mehrerer Deep-Neural-Networks (DNNs)), durchführen. Die resultierende Darstellung der 3D-Oberflächenstruktur (z. B. dichte Erkennungsdaten 120) kann jede geeignete Form annehmen, wie z. B. eine 2D-Höhenkarte und/oder eine 3D-Punktwolke.

[0041] **Fig. 3** ist ein Diagramm, das eine beispielhafte Implementierung des Erkennungsverdichters 115 gemäß einigen Ausführungsformen der vorliegenden Offenbarung zeigt. In **Fig. 3** umfasst der Erkennungsverdichter 115 einen Markov-Zufallsfeld-Oberflächenschätzer 310 und einen Deep-Learning-Modell-Oberflächenschätzer 320.

[0042] In einigen Ausführungsformen kann der Markov-Zufallsfeld-Oberflächenschätzer 310 die spärlichen Erfassungsdaten 110 verdichten, um eine dichtere Darstellung der 3D-Oberflächenstruktur (z. B. die dichten Erfassungsdaten 120) zu erzeugen. Zum Beispiel kann der Markov-Zufallsfeld-Oberflächenschätzer 310 eine spärliche 2D-Top-Down-Höhenkarte o (oder ein anderes spärliches Projektionsbild) verdichten, indem er eine dichte 2D-Top-Down-Höhenkarte g (oder ein anderes dichtes Projektionsbild) aus o ableitet. Genauer gesagt kann die Beziehung zwischen g und o mit einem probabilistischen Modell wie einem Markov-Zufallsfeld modelliert werden, und der Markov-Zufallsfeld-Oberflächenschätzer 310 kann eine Maximum-a-Posterior (MAP)-Inferenz durchführen, um das wahrscheinlichste g angesichts des probabilistischen Modells und eines Satzes von beobachteten Werten o zu schätzen, ein ungerichteter Graph) als probabilistisches Modell verwendet werden, da es in der Lage ist, räumliche Abhängigkeiten zu modellieren, wie sie in bestimmten 3D-Oberflächenstrukturen, z. B. 3D-Straßenoberflächen, bestehen, bei denen lokale Oberflächenbereiche oft glatt sind. So kann in einigen Ausführungsformen die Beziehung zwischen einer spärlichen Höhenkarte o und einer dichten Höhenkarte g mit einem ungerichteten Graphen modelliert werden.

[0043] **Fig. 4** ist ein Diagramm, das einen beispielhaften ungerichteten Graphen 400 zeigt, der die Beziehung zwischen einer spärlichen Höhenkarte o und einer dichten Höhenkarte g gemäß einigen Ausführungsformen der vorliegenden Offenbarung modelliert. Zum Beispiel kann jedes Pixel in der dichten Höhenkarte g mit einem entsprechenden Knoten modelliert werden (z. B. g_1, g_2, g_3, g_4, in **Fig. 4**), der Kanten hat, die mit benachbarten Knoten verbunden sind (z. B. eine Kante für jedes benachbarte Pixel). In einigen Ausführungsformen können Knoten, die inneren Pixeln in g entsprechen, vier Kanten (z. B. Verbindungen zu horizontal und vertikal benachbarten Knoten), acht Kanten (z. B. Verbindungen zu horizontal, vertikal und diagonal benachbarten Knoten) oder andere aufweisen. **Fig. 4** zeigt einen Teil eines ungerichteten Graphen, der den inneren Pixeln in g entspricht, wobei jeder entsprechende Knoten g_1, g_2, g_3, g_4 vier Kanten aufweist. Knoten, die Randpixeln in g entsprechen, können drei Kanten (z. B. Verbindung zu horizontal und vertikal benachbarten Knoten), fünf Kanten (z. B. Verbindung zu horizontal, vertikal und diagonal benachbarten Knoten) oder andere aufweisen. Knoten, die Eckpixeln in g entsprechen, können zwei Kanten (z. B. Verbindung zu horizontal und vertikal benachbarten Knoten), drei Kanten (z. B. Verbindung zu horizontal, vertikal und diagonal benachbarten Knoten) oder andere aufweisen.

[0044] Außerdem kann jedes Pixel in der spärlichen Höhenkarte o als verrauschte Beobachtung eines entsprechenden Pixels in der dichten Höhenkarte g betrachtet werden. Somit kann jedes Pixel in der spärlichen Höhenkarte o mit einem Knoten modelliert werden, der eine Kante hat, die mit einem entsprechenden Knoten (der das entsprechende Pixel darstellt) aus der dichten Höhenkarte g verbunden ist. **Fig. 4** zeigt einen Teil eines ungerichteten Graphen mit den Knoten o_1, o_2, o_3, o_4 (die Pixel in der spärlichen Höhenkarte o darstellen), die mit den Knoten g_1, g_2, g_3, g_4 (die Pixel in der dichten Höhenkarte g darstellen) verbunden sind.

[0045] Anders ausgedrückt kann in einigen Ausführungsformen eine zu modellierende Oberfläche aus einer gewünschten Perspektive (z. B. von oben nach unten) betrachtet und in ein 2D-Gitter unterteilt werden, und es kann ein ungerichteter Graph mit einem 3D-Gitter mit zwei Knotenschichten gebildet werden, wobei jede Schicht einen Knoten für jede Zelle oder jeden Schnittpunkt im 2D-Gitter aufweist. Die untere Schicht kann der Ground Truth entsprechen (z. B. die dichte Höhenkarte g), und die obere Schicht kann einer verrauschten Beobachtung entsprechen (die spärliche Höhenkarte o). Beachten Sie, dass die Schicht, die einer verrauschten Beobachtung entspricht, einen Knoten für jede Zelle oder jeden Schnittpunkt im 2D-Gitter enthalten kann, auch wenn die verrauschte Beobachtung spärlich sein kann. Daher haben einige Knoten, die Pixeln in der spärlichen Höhenkarte o entsprechen, möglicherweise keine entsprechenden Beobachtungen.

[0046] Nachdem die Beziehung zwischen einer spärlichen Höhenkarte o und einer dichten Höhenkarte g mit einem Markov-Zufallsfeld (z. B. einem ungerichteten Graphen) modelliert wurde, kann jeder Knoten in dem Modell als Zufallsvariable betrachtet werden, so dass die gemeinsame Wahrscheinlichkeitsverteilung aller Zufallsvariablen wie folgt geschrieben werden kann:

$$P(g,o) = \frac{1}{z}\prod_{i,j} \quad \psi\left(g_i, g_j\right)\prod_i \quad \phi(g_i, o_i),$$

(Gleichung 1)

wobei ψ (g_i, g_j) ein paarweiser Potentialterm ist, der die Höhenbeziehung zwischen benachbarten Pixeln in g darstellt, ϕ(g_i, o_i) ein unärer Potentialterm ist, der die Beziehung zwischen der wahren Höhe g_i und der beobachteten verrauschten Höhe o_i angibt, und Z eine Normierungskonstante ist, die sicherstellt, dass die Summe der Komponentenverteilungen eins ergibt.

[0047] Um besondere Abhängigkeiten zwischen benachbarten Pixeln in g darzustellen, kann der paarweise Potentialterm in einigen Ausführungsformen die folgende Form annehmen:

$$\psi\left(g_i, g_j\right) = exp\left(-w_{ij}\left(g_i - g_j\right)^2\right), \qquad \text{(Gl. 2)}$$

wobei w_{ij} das Gewicht zwischen den Knotenpunkten (g_i, g_j) angibt, die benachbarten Pixeln entsprechen, wie im Folgenden näher erläutert wird. Um einen Beitrag von beobachteten Pixeln darzustellen, kann man annehmen, dass o eine verrauschte Version von g ist:

$o_i = g_i + Rauschen$, wenn *pixel i* beobachtet wird. (Gl. 3)

In einigen Ausführungsformen kann der unäre Potentialterm daher wie folgt angegeben werden:

$$\phi\left(g_i, o_i\right) = exp\left(-c_i\left(g_i - o_i\right)^2\right), \qquad \text{(Gl. 4)}$$

wobei c_i ein Gewicht für Pixel i angibt und c_i auf 0 gesetzt werden kann, wenn Pixel i nicht beobachtet wird. Im Allgemeinen können beliebige geeignete Gewichtungen für w_{ij} und c_i gewählt werden, um z. B. den paarweisen Potentialterm (z. B. zur Hervorhebung von Beziehungen zwischen benachbarten Pixeln) oder den unären Potentialterm (z. B. wenn ein relativ hohes Vertrauen in die beobachteten Werte besteht) stärker zu betonen. In einigen Ausführungsformen kann ein gemeinsames Gewicht für alle Paare ausgewählt werden w_{ij} ein gemeinsames Gewicht für jedes Paar ausgewählt werden, das c_i entsprechend einem beobachteten Pixel ausgewählt werden, ein Hyperparameter kann für jedes Gewicht ausgewählt werden, um ein gewünschtes Verhältnis zwischen w_{ij} und c_i zu bilden, und/oder auf andere Weise.

[0048] Mit der gemeinsamen Wahrscheinlichkeitsverteilung P (g, o) und einem Satz beobachteter Werte der dünnbesetzten Höhenkarte o (oder anderer dünnbesetzter Erkennungsdaten 110) kann der Markov-Zufallsfeld-Oberflächenschätzer 310 einen Wert (z. B. eine Höhenschätzung) für jedes Pixel in der dichten Höhenkarte g (oder andere dichte Erkennungsdaten 120) vorhersagen, indem er einen beliebigen bekannten MAP-Inferenzalgorithmus verwendet, wie z. B. Iterative Conditional Mode, Gaussian Belief Propagation oder andere. Im Allgemeinen kann der Markov-Zufallsfeld-Oberflächenschätzer 310 eine dichte Darstellung g einer 3D-Oberflächenstruktur aus einer spärlichen Darstellung o (z. B. eine verrauschte und partielle Beobachtung) der 3D-Oberflächenstruktur schätzen. Das Ergebnis kann eine Darstellung der 3D-Oberflächenstruktur der Straße sein, wie z. B. eine 2D-Höhenkarte, die in eine 3D-Punktwolke (z. B. in 3D-Weltkoordinaten) transformiert werden kann. Im Betrieb kann der Markov-Zufallsfeld-Oberflächenschätzer 310 wiederholt auf aufeinanderfolgende Instanzen der spärlichen Erfassungsdaten 110 (z. B. abgeleitet von Sensordaten, die während aufeinanderfolgender Zeitscheiben erfasst wurden, die durch ein bestimmtes internes Element getrennt sind) einwirken, um aufeinanderfolgende Instanzen der dichten Erfassungsdaten 120 (z. B. aufeinanderfolgende Darstellungen entsprechender Abschnitte der 3D-Oberflächenstruktur der Straße) vorherzusagen, wenn sich das Fahrzeug 1700 der **Fig. 17A-17D** durch die 3D-Umgebung bewegt.

[0049] Zusätzlich oder alternativ kann der Deep-Leaming-Modell-Oberflächenschätzer 320 die Darstellung der 3D-Oberflächenstruktur verdichten.

Zum Beispiel kann der Deep-Learning-Modell-Oberflächenschätzer 320 die spärlichen Erfassungsdaten 110 (z. B. eine spärliche 2D-Top-Down-Höhenkarte o) verdichten, indem er Werte der dichten Erfassungsdaten 120 (z. B. eine dichte 2D-Top-Down-Höhenkarte g) aus den spärlichen Erfassungsdaten 110 unter Verwendung eines oder mehrerer Deep-Learning-Modelle ableitet. Von daher kann der Deep-Learning-Modell-Oberflächenschätzer 320 die Beziehung zwischen den spärlichen Erfassungsdaten 110 (z. B. eine Darstellung spärlicher und verrauschter Beobachtungen, wie ein Projektionsbild einer 3D-Punktwolke) und den dichten Erfassungsdaten 120 (z. B. eine dichtere Darstellung der 3D-Oberflächenstruktur, wie ein Projektionsbild einer dichten 3D-Punktwolke) erlernen.

[0050] Fig. 5 ist ein Datenflussdiagramm, das eine beispielhafte Implementierung des Deep-Learning-Modell-Oberflächenschätzers 320 gemäß einigen Ausführungsformen der vorliegenden Offenbarung veranschaulicht. Auf einer hohen Ebene kann der Deep-Learning-Modell-Oberflächenschätzer 320 einen Vorprozessor 510, ein oder mehrere Deep-Learning-Modell(e) 535, die konfiguriert sind, Werte der dichten Erfassungsdaten 120 vorherzusagen, und einen Nachprozessor 575 umfassen. Der Vorprozessor 510 kann die spärlichen Erkennungsdaten 110 in Eingabedaten 530 codieren, die das/die Deep-Learning-Modell(e) 535 unterstützen, und die Eingabedaten 530 können in das/die Deep-Learning-Modell(e) 535 eingespeist werden, um Regressionsdaten 570 und/oder Konfidenzdaten 580 vorherzusagen. In einigen Ausführungsformen können die Regressionsdaten 570 und/oder die Konfidenzdaten 580, die von dem/den Deep-Learning-Modell(en) 535 vorhergesagt wurden, als die Dichtheitserfassungsdaten 120 verwendet werden. In einigen Ausführungsformen kann der Vorprozessor 510 einen Normalisator 520 enthalten, der einen systematischen Fehler aus den Eingabedaten 530 entfernt. In diesem Fall kann der Nachprozessor 575 den systematischen Fehler wieder in die Regressionsdaten 570 einbringen, die von dem/den Deep-Learning-Modell(en) 535 vorhergesagt wurden, um mindestens einen Teil der dichten Erkennungsdaten 120 zu erzeugen.

[0051] In einigen Ausführungsformen umfasst der Vorprozessor 510 einen Encoder 515, der die spärlichen Erkennungsdaten 110 in eine Darstellung codiert, die von dem/den Deep-Learning-Modell(en) 535 unterstützt wird. Als nicht einschränkendes Beispiel kann der Encoder 515 in einigen Ausführungsformen, in denen die spärlichen Erkennungsdaten 110 eine spärliche 3D-Punktwolke enthalten, die spärliche 3D-Punktwolke projizieren, um ein spärliches Projektionsbild zu bilden (z. B. eine Top-Down-Höhenkarte). In einigen Fällen (z. B. ohne den Normalisator 520) kann das resultierende spärliche Projektionsbild als Eingabedaten 530 verwendet und in das/die Deep-Learning-Modell(e) 535 eingespeist werden, um die Regressionsdaten 570 (z. B. ein dichtes Projektionsbild wie eine Top-Down-Höhenkarte) und/oder die Konfidenzdaten 580 vorherzusagen. In einigen Fällen können die von dem/den Deep-Leaming-Modell(en) 535 vorhergesagten Regressionsdaten 570 und/oder die Konfidenzdaten 580 als die dichten Erkennungsdaten 120 verwendet werden.

[0052] In einigen Fällen können die spärlichen Erkennungsdaten 110 und/oder die codierten spärlichen Erkennungsdaten (z. B. ein spärliches Projektionsbild) einen systematischen Fehler enthalten. Von daher kann der Vorprozessor 510 in einigen Ausführungsformen einen Normalisator 520 enthalten, der den systematischen Fehler entfernt oder die spärlichen Erkennungsdaten 110 und/oder die codierten spärlichen Erkennungsdaten anderweitig normalisiert. Beispielsweise kann in einigen Ausführungsformen, in denen die spärlichen Erkennungsdaten 110 eine spärliche 3D-Punktwolke enthalten und der Encoder 515 die spärliche 3D-Punktwolke projiziert, um eine 2D-Höhenkarte zu bilden, die 2D-Höhenkarte Höhenwerte speichern, die einen systematischen Fehler enthalten, die der Höhe der Kamera entspricht, die die Bilder aufgenommen hat, aus denen die 2D-Höhenkarte abgeleitet wurde. In einigen Ausführungsformen berechnet der Normalisator 520 die mittlere Höhe der Höhenwerte (z. B. einer gewünschten Oberfläche) in der 2D-Höhenkarte und subtrahiert die mittlere Höhe von allen Höhenwerten (z. B. der gewünschten Oberfläche), um den systematischen Fehler zu entfernen. Die resultierende 2D-Höhenkarte (oder andere normalisierte, codierte spärliche Erkennungsdaten) kann als Eingabedaten 530 verwendet und in das/die Deep-Learning-Modell(e) 535 eingespeist werden, um die Regressionsdaten 570 (z. B. eine dichte 2D-Höhenkarte) und/oder die Konfidenzdaten 580 vorherzusagen, und der Nachprozessor 575 kann den systematischen Fehler in die vorhergesagte Ausgabe wieder einführen (z. B. durch Hinzufügen des systematischen Fehlers zu einigen oder allen vorhergesagten Werten der Regressionsdaten 570).

[0053] Sich nun den Deep-Learning-Modellen 535 zuwendend, so können das/die Deep-Learning-Modell(e) 535 in einigen Ausführungsformen unter Verwendung eines DNN, wie z. B. eines neuronalen Faltungsnetz (CNN), implementiert werden. Obwohl bestimmte Ausführungsformen beschrieben werden, bei denen das/die Deep-Learning-Modell(e) 535 unter Verwendung von neuronalen Netzwerken und insbesondere CNN(s) implementiert wird/werden, ist dies nicht als Einschränkung zu verstehen. Zum Beispiel und ohne Einschränkung können das (die) Deep-Learning-Modell(e) 535 zusätzlich oder alternativ jede Art von maschinellem Lernmodell umfas-

sen, wie z. B. ein maschinelles Lernmodell (maschinelle Lernmodelle) unter Verwendung von linearer Regression, logistischer Regression, Entscheidungsbäumen, Support-Vektor-Maschinen (SVM), Naive Bayes, k-nearest neighbor (Knn), K-Mittel-Clustering, Random Forest, Dimensionalitätsreduktionsalgorithmen, Gradient-Boosting-Algorithmen, Markov-Zufallsfelder, neuronalen Netzen (z. B., Autoencoder, Faltungsalgorithmen, rekurrente Algorithmen, Perzeptronen, Long/Short Term Memory (LSTM), Hopfield, Boltzmann, Deep Belief, Deconvolutional, Generative Adversarial, Liquid State Machine usw.) und/oder andere Arten von maschinellen Lemmodellen.

[0054] In einigen Ausführungsformen können die Deep-Learning-Modelle 535 einen gemeinsamen Stamm (oder einen Strom von Schichten) mit einem oder mehreren Köpfen (oder mindestens teilweise diskreten Strömen von Schichten) zur Vorhersage verschiedener Ausgaben basierend auf der Eingabedaten 530 enthalten. Beispielsweise können die Deep-Learning-Modelle 535 ohne Einschränkung einen Merkmalsextraktor (z. B. einen DNN, einen Encoder/Decoder usw.) mit Faltungsschichten, Pooling-Schichten und/oder anderen Schichttypen umfassen, wobei die Ausgabe des Merkmalsextraktors als Eingabe für jeden einer Vielzahl von Köpfen bereitgestellt wird, die unterschiedliche Ausgaben vorhersagen. Die verschiedenen Köpfe können in einigen Beispielen parallele Eingaben erhalten und somit unterschiedliche Ausgaben aus ähnlichen Eingabedaten erzeugen. Im Beispiel der **Fig. 5** wird das Deep-Learning-Modell 535 mit einer beispielhaften Architektur dargestellt, die Merkmale aus den Eingabedaten 106 extrahiert und eine Regression an den extrahierten Merkmalen durchführt. Genauer gesagt können die Deep-Learning-Modelle 535 einen Encoder/Decoder 540, einen Regressionskopf 545 und/oder einen Konfidenzkopf 550 umfassen.

[0055] Der Encoder/Decoder 540 kann unter Verwendung von Encoder- und Decoderkomponenten mit Sprungverbindungen implementiert werden (z. B. ähnlich wie ResNet, Feature Pyramid Network, U-Net usw.). Beispielsweise kann der Encoder/Decoder 540 die Eingabedaten 530 (z. B. ein Projektionsbild, eine 2D-Höhenkarte) akzeptieren und verschiedene Faltungen, Pooling und/oder andere Arten von Operationen anwenden, um Merkmale in einen latenten Raum zu extrahieren. In **Fig. 5** ist der Encoder/Decoder 540 mit einer Beispielimplementierung dargestellt, die (von links nach rechts) einen codierenden (kontrahierenden) Weg und einen decodierenden (expansiven) Weg umfasst. Entlang des kontrahierenden Wegs kann jede Auflösung eine beliebige Anzahl von Schichten (z. B. Faltungen, erweiterte Faltungen, Inception-Blöcke usw.) und eine Downsampling-Operation (z. B. Max-Pooling) umfassen. Auf dem expansiven Weg kann jede Auflösung eine beliebige Anzahl von Schichten umfassen (z. B. Enfaltungen, Upsampling gefolgt von Faltung(en) und/oder andere Arten von Operationen). Im expansiven Weg kann jede Auflösung einer Merkmalskarte hochabgetastet und (z. B. in der Tiefendimension) mit Merkmalskarten der gleichen Auflösung aus dem kontrahierenden Weg verkettet werden. In diesem Beispiel können entsprechende Auflösungen des kontrahierenden und des expansiven Wegs mit Sprungverbindungen verbunden werden, die zum Hinzufügen oder Verketten von Merkmalskarten mit entsprechenden Auflösungen verwendet werden können. Von daher kann der Encoder/Decoder 540 Merkmale in einen latenten Raum extrahieren, und eine Darstellung der extrahierten Merkmale kann in den Regressionskopf 545 und/oder den Konfidenzkopf 550 eingegeben werden.

[0056] Der Regressionskopf 545 kann eine beliebige Anzahl von Schichten (z. B. Faltungen, Pooling, Klassifizierer wie Softmax und/oder andere Arten von Operationen usw.) enthalten, die eine bestimmte Art von Informationen über die interessierenden 3D-Oberflächenstruktur (z. B. einen Höhenwert für jedes Pixel) aus der Ausgabe des Encoders/Decoders 540 vorhersagen. In einigen Ausführungsformen können die vom Regressionskopf 545 vorhergesagten Regressionsdaten 570 die Form einer 2D-Höhenkarte annehmen, wobei jedes Pixel eine Gleitkommazahl speichert, die die Höhe des durch das Pixel dargestellten Teils der 3D-Oberfläche regressiert.

[0057] Der Konfidenzkopf 550 kann eine beliebige Anzahl von Schichten (z. B. Faltungen, Pooling, Klassifizierer wie Softmax und/oder andere Arten von Operationen usw.) enthalten, die die Konfidenzdaten 580 für die Regressionsdaten 570 aus der Ausgabe des Encoders/Decoders 540 vorhersagen. Zum Beispiel können in einigen Ausführungsformen, in denen die Regressionsdaten 570 die Form einer 2D-Höhenkarte $\in N^{m \times n}$ annehmen, können die Konfidenzdaten 580 die Form einer entsprechenden Konfidenzkarte $\in N^{m \times n}$ mit Pixeln annehmen, die eine Gleitkommazahl speichern, die eine Darstellung der Konfidenz eines entsprechenden vorhergesagten Wertes in der 2D-Höhenkarte regressiert.

[0058] So zeigt **Fig. 5** eine Ausführungsform des Deep-Learning-Modells/der Deep-Learning-Modelle 535, das Regressionsdaten 570 (z. B. eine 2D-Höhenkarte) und Konfidenzdaten 580 vorhersagt. Es kann jedoch eine beliebige Anzahl von Variationen implementiert werden. In einigen Ausführungsformen kann das/die Deep-Learning-Modell(e) 535 beispielsweise mit einem einzigen Ausgangskanal implementiert werden, der dem Regressionskopf 545 entspricht. In einem anderen Beispiel kann das/die Deep-Learning-Modell(e) 535 in einigen Ausführungsformen eine oder mehrere rekurrente Schich-

ten (z. B. Gated Recurrent Units, Long Short Term Memory) enthalten, um zeitliche Informationen zu nutzen. In einigen Fällen kann die Einbeziehung einer oder mehrerer rekurrenter Schichten es dem/den Deep-Learning-Modell(en) 535 ermöglichen, Informationen aus früheren Zeitabschnitten zu nutzen, was zu besseren Vorhersagen und stabileren Verdichtungsergebnissen im Zeitverlauf führt. In einem weiteren Beispiel können das/die Deep-Learning-Modell(e) 535 mit mehreren Eingabeköpfen implementiert werden, die unterschiedliche Eingaben akzeptieren, wie ein Eingabebild (z. B. ein RGB-Bild) mit einer perspektivischen Ansicht und ein Projektionsbild mit einer anderen Ansicht (z. B. eine Top-Down-Höhenkarte). **Fig. 6** stellt ein solches Beispiel dar.

[0059] Genauer gesagt ist **Fig. 6** ein Datenflussdiagramm, das eine Beispielimplementierung des Deep-Learning-Modell-Oberflächenschätzers 320 der **Fig. 3** mit einem Deep-Learning-Modell bzw. mehreren Deep-Learning-Modellen zeigt, die mehrere Eingabeköpfe enthalten. Im Allgemeinen weisen die in den **Fig. 5** und **Fig. 6** dargestellten Implementierungen des Deep-Learning-Modell-Oberflächenschätzers 320 im Allgemeinen ähnliche Komponenten auf, außer dass die in **Fig. 6** dargestellte Implementierung das/die Deep-Learning-Modell(e) 535 um einen Bildencoder 610 erweitert. Während die in **Fig. 5** dargestellte Implementierung des Deep-Learning-Modells 535 einen einzelnen Eingangskopf (z. B. den Encoderteil des Encoders/Decoders 540) umfasst, der die Eingangsdaten 530 (z. B. eine Projektion einer 3D-Punktwolke) akzeptiert, akzeptiert die in **Fig. 6** dargestellte Implementierung des Deep-Learning-Modells 535 zusätzlich die Bilddaten 102 (z. B. einen RBG-Frame) in einem zweiten Eingangskopf (z. B. den Bildencoder 610). Im Allgemeinen kann der Bildencoder 610 (und/oder jeder andere Eingabekopf) eine beliebige Anzahl von Schichten (z. B. Faltungen, Pooling und/oder andere Arten von Operationen) enthalten, um Merkmale in einen latenten Raum zu extrahieren, und die extrahierten Merkmale können mit extrahierten Merkmalen aus dem Encoderteil des Encoders/Decoders 540 (und/oder extrahierten Merkmalen aus anderen Eingabeköpfen) kombiniert (z. B. verkettet) werden. So können in einigen Ausführungsformen wie der in **Fig. 6** dargestellten die Deep-Learning-Modelle 535 aus zwei verschiedenen Ansichten einer beobachteten Oberflächenstruktur lernen (z. B. Top-Down und Perspektive, 3D-Punktwolkenraum und 2D-Bildraum usw.).

[0060] Von daher und um auf **Fig. 3** zurückzukommen, kann der Deep-Learning-Modell-Oberflächenschätzer 320 der **Fig. 3** unter Verwendung einer Vielzahl von Architekturen für ein konstituierendes Deep-Learning-Modell (z. B. das/die Deep-Learning-Modell(e) 535 der **Fig. 5** oder **Fig. 6**) und/oder ein anderes maschinelles Lernmodell bzw. andere maschinelle Lernmodelle implementiert werden, um die dichten Erfassungsdaten 120 aus den spärlichen Erfassungsdaten 110 vorherzusagen. Das Ergebnis kann eine Darstellung der 3D-Oberflächenstruktur der Straße sein, wie z. B. eine 2D-Höhenkarte, die in eine 3D-Punktwolke transformiert werden kann (z. B. in 3D-Weltkoordinaten). Im Betrieb kann der Deep-Learning-Modell-Oberflächenschätzer 320 wiederholt auf aufeinanderfolgende Instanzen der spärlichen Erkennungsdaten 110 (z. B. abgeleitet von Sensordaten, die während aufeinanderfolgender Zeitscheiben erfasst wurden, die durch einen bestimmten internen Bereich getrennt sind) einwirken, um aufeinanderfolgende Instanzen der dichten Erkennungsdaten 120 (z. B. aufeinanderfolgende Darstellungen entsprechender Teile der 3D-Oberflächenstruktur der Straße) vorherzusagen, wenn sich beispielsweise das Fahrzeug 1700 der **Fig. 17A-17D** durch die 3D-Umgebung bewegt.

[0061] Auf **Fig. 1** zurückkommend: Sobald die 3D-Struktur der erfassten Oberfläche bestimmt wurde, können Positionswerte, die nicht bereits in 3D-Weltkoordinaten vorliegen, in 3D-Weltkoordinaten umgewandelt, mit einer entsprechenden Klassenkennzeichnung, das die erfasste Oberfläche (z. B. eine Straße) identifiziert, verknüpft und/oder zur Verwendung durch das Fahrzeug 1700 der **Fig. 17A-17D** bei der Durchführung einer oder mehrerer Operationen bereitgestellt werden. Zum Beispiel können die dichten Erkennungsdaten 120 (z. B. eine 3D-Punktwolke, ein Projektionsbild, entsprechende Beschriftungen) von Steuerkomponenten des Fahrzeugs 1700 verwendet werden, wie z. B. einem Software-Stapel 122 für autonomes Fahren, der auf einer oder mehreren Komponenten des Fahrzeugs 1700 der **Fig. 17A-17D** (z. B. der/die SoC(s) 1704, die CPU(s) 1718, die GPU(s) 1720 usw.). Zum Beispiel kann das Fahrzeug 1700 diese Informationen (z. B. Instanzen von Hindernissen) verwenden, um zu navigieren, zu planen oder anderweitig eine oder mehrere Operationen (z. B. Hindernis- oder Höckervermeidung, Spurhaltung, Spurwechsel, Zusammenführen, Aufteilen, Anpassen eines Aufhängungssystems des Ego-Objekts oder Ego-Aktors an die aktuelle Straßenoberfläche, Anwenden einer frühen Beschleunigung oder Abbremsung basierend auf einer sich nähernden Oberflächenneigung, Kartierung usw.) innerhalb der Umgebung durchzuführen.

[0062] In einigen Ausführungsformen können die dichten Erkennungsdaten 120 von einer oder mehreren Schichten des Software-Stapels für autonomes Fahren 122 (hier alternativ als „Fahrstapel 122“ bezeichnet) verwendet werden. Der Fahrstapel 122 kann einen Sensormanager (nicht dargestellt), Wahrnehmungskomponente(n) (z. B. entsprechend einer Wahrnehmungsschicht des Fahrstapels 122), einen Weltmodellmanager 126, Planungskomponente(n)

128 (z. B. entsprechend einer Planungsschicht des Fahrstapels 122), Steuerkomponente(n) 130 (z. B., (z. B. entsprechend einer Steuerungsschicht des Fahrstapels 122), Hindernisvermeidungskomponente(n) 132 (z. B. entsprechend einer Hindernis- oder Kollisionsvermeidungsschicht des Fahrstapels 122), Betätigungskomponente(n) 134 (z. B. entsprechend einer Betätigungsschicht des Fahrstapels 122) und/oder andere Komponenten, die zusätzlichen und/oder alternativen Schichten des Fahrstapels 122 entsprechen. Der Prozess 100 kann in einigen Beispielen von der/den Wahrnehmungskomponente(n) ausgeführt werden, die die Schichten des Fahrstapels 122 an den Weltmodellmanager weiterleiten können, wie hierin ausführlicher beschrieben.

[0063] Der Sensormanager kann Sensordaten von den Sensoren des Fahrzeugs 1700 verwalten und/oder abstrahieren. Zum Beispiel können die Sensordaten (z. B. ständig, in Intervallen, basierend auf bestimmten Bedingungen) von RADAR-Sensor(en) 1760 erzeugt werden (siehe **Fig. 17C**). Der Sensormanager kann die Sensordaten von den Sensoren in verschiedenen Formaten empfangen (z. B. können Sensoren desselben Typs Sensordaten in verschiedenen Formaten ausgeben) und kann so konfiguriert sein, dass er die verschiedenen Formate in ein einheitliches Format umwandelt (z. B. für jeden Sensor desselben Typs). Infolgedessen können andere Komponenten, Merkmale und/oder Funktionen des autonomen Fahrzeugs 1700 das einheitliche Format verwenden, wodurch die Verarbeitung der Sensordaten vereinfacht wird. In einigen Beispielen kann der Sensormanager ein einheitliches Format verwenden, um die Sensoren des Fahrzeugs 1700 zu steuern, z. B. um Bildraten einzustellen oder eine Verstärkungsregelung durchzuführen. Der Sensormanager kann auch Sensorpakete oder Kommunikationen, die den Sensordaten entsprechen, mit Zeitstempeln aktualisieren, um die Verarbeitung der Sensordaten durch verschiedene Komponenten, Merkmale und Funktionen eines autonomen Fahrzeugsteuerungssystems zu unterstützen.

[0064] Ein Weltmodell-Manager 126 kann zur Erzeugung, Aktualisierung und/oder Definition eines Weltmodells verwendet werden. Der Weltmodell-Manager 126 kann Informationen verwenden, die von der (den) Wahrnehmungskomponente(n) des Fahrstapels 122 erzeugt und empfangen werden (z. B. die Positionen der erkannten Hindernisse). Die Wahrnehmungskomponente(n) können eine Hinderniswahrnehmung, eine Wegwahmehmung, eine Wartewahmehmung, eine Kartenwahrnehmung und/oder eine andere Wahrnehmungskomponente (n) umfassen. Beispielsweise kann das Weltmodell mindestens teilweise basierend auf Affordanzen für Hindernisse, Wege und Wartebedingungen definiert werden, die in Echtzeit oder nahezu in Echtzeit von dem Hinderniswahrnehmer, dem Wegwahrnehmer, dem Wartewahrnehmer und/oder dem Kartenwahmehmer wahrgenommen werden können. Der Weltmodellmanager 126 kann das Weltmodell basierend auf neu erzeugten und/oder empfangenen Eingaben (z. B. Daten) von dem Hinderniswahrnehmer, dem Wegwahrnehmer, dem Wartewahrnehmer, dem Kartenwahrnehmer und/oder anderen Komponenten des autonomen Fahrzeugsteuerungssystems kontinuierlich aktualisieren.

[0065] Das Weltmodell kann verwendet werden, um die Planungskomponente(n) 128, die Steuerkomponente(n) 130, die Hindernisvermeidungskomponente (n) 132 und/oder die Betätigungskomponente(n) 134 des Fahrstapels 122 zu informieren. Der Hinderniswahrnehmer kann eine Hinderniswahrnehmung durchführen, die darauf beruhen kann, wo das Fahrzeug 1700 fahren darf oder fahren kann (z. B. basierend auf dem Standort der befahrbaren oder anderen navigierbaren Wege, die durch die Vermeidung erkannter Hindernisse in der Umgebung und/oder erkannter Erhebungen in der Straßenoberfläche definiert sind), und wie schnell das Fahrzeug 1700 fahren kann, ohne mit einem Hindernis zu kollidieren (z. B. einem Objekt, wie einer Struktur, einer Einheit, einem Fahrzeug usw.), das von den Sensoren des Fahrzeugs 1700 erfasst wird.

[0066] Der Wegwahrnehmer kann eine Wegwahrnehmung durchführen, indem er z. B. nominelle Wege wahrnimmt, die in einer bestimmten Situation verfügbar sind. In einigen Beispielen kann der Wegwahrnehmer außerdem Fahrspurwechsel für die Wegwahrnehmung in Betracht ziehen. Ein Fahrspurgraph kann den Weg oder die Wege darstellen, die dem Fahrzeug 1700 zur Verfügung stehen, und kann so einfach sein wie ein einzelner Weg auf einer Autobahnauffahrt. In einigen Beispielen kann der Fahrspurgraph Wege zu einer gewünschten Fahrspur enthalten und/oder verfügbare Änderungen auf der Autobahn (oder einem anderen Straßentyp) anzeigen, oder er kann nahe gelegene Fahrspuren, Fahrspuränderungen, Gabelungen, Abbiegungen, Kleeblattkreuzungen, Zusammenführungen und/oder andere Informationen enthalten. In einigen Ausführungsformen kann der Wegwahrnehmer die dichten Erkennungsdaten 120 berücksichtigen. Beispielsweise kann der Wegwahrnehmer eine rekonstruierte 3D-Straßenoberfläche auswerten, um Erhebungen zu identifizieren und Wege einzubeziehen, die diese Erhebungen vermeiden.

[0067] Der Wertewahrnehmer kann dafür verantwortlich sein, Einschränkungen für das Fahrzeug 1700 zu bestimmen, die sich aus Regeln, Konventionen und/oder praktischen Erwägungen ergeben. Die Regeln, Konventionen und/oder praktischen Erwägungen können sich beispielsweise auf eine 3D-Straßenoberfläche, Ampeln, mehrspurige Haltestel-

len, Abzweigungen, Einmündungen, Mautstellen, Schranken, Polizei oder andere Einsatzkräfte, Straßenarbeiter, angehaltene Busse oder andere Fahrzeuge, einspurige Brückenübergänge, Fähreneinfahrten usw. beziehen. Auf diese Weise kann der Wertewahrnehmer genutzt werden, um potentielle Hindernisse zu erkennen und eine oder mehrere Steuerungen durchzuführen (z. B. Abbremsen, Anhalten usw.), die allein mit dem Hinderniswahrnehmer nicht möglich gewesen wären. In einigen Ausführungsformen kann der Wertewahrnehmer die dichten Erkennungsdaten 120 berücksichtigen. Beispielsweise kann der Wertewahrnehmer eine rekonstruierte 3D-Straßenoberfläche auswerten, um eine sich nähernde Oberflächenneigung zu identifizieren und zu bestimmen, dass eine frühzeitige Beschleunigung oder Abbremsung angewendet wird, um sich der sich nähernden Oberflächenneigung anzupassen. Zusätzlich oder alternativ kann der Wartewahrnehmer eine rekonstruierte 3D-Straßenoberfläche auswerten, um einen Abschnitt einer sich nähernden Straßenoberfläche zu identifizieren, und bestimmen, ein Aufhängungssystem des Fahrzeugs 1700 so anzupassen und/oder anzupassen, dass das Aufhängungssystem an die identifizierte Straßenoberfläche angepasst wird, sobald das Fahrzeug 1700 einen entsprechenden Abschnitt der Straße erreicht.

[0068] Der Kartenwahrnehmer kann einen Mechanismus enthalten, mit dem Verhaltensweisen erkannt werden, und in einigen Beispielen spezifische Beispiele dafür bestimmen, welche Konventionen an einem bestimmten Ort angewendet werden. Beispielsweise kann der Kartenwahrnehmer anhand von Daten, die frühere Fahrten oder Fahrten darstellen, feststellen, dass an einer bestimmten Kreuzung zwischen bestimmten Uhrzeiten keine Kehrtwendungen erlaubt sind, dass ein elektronisches Schild, das die Richtung der Fahrspuren anzeigt, sich je nach Tageszeit ändert, dass zwei Ampeln in unmittelbarer Nähe (z. B. nur geringfügig versetzt) verschiedenen Straßen zugeordnet sind, dass in Rhode Island das erste Auto, das an einer Ampel auf das Linksabbiegen wartet, gegen das Gesetz verstößt, indem es vor dem Gegenverkehr abbiegt, wenn die Ampel auf Grün schaltet, und/oder andere Informationen. Der Kartenwahrnehmer kann das Fahrzeug 1700 über statische oder stationäre Infrastrukturobj ekte und Hindernisse informieren. Der Kartenwahrnehmer kann auch Informationen für den Wartewahrnehmer und/oder den Wegwahrnehmer generieren, z. B. um zu bestimmen, welche Ampel an einer Kreuzung grün sein muss, damit das Fahrzeug 1700 einen bestimmten Weg einschlagen kann.

[0069] In einigen Beispielen können Informationen vom Kartenwahrnehmer an einen oder mehrere Server gesendet, übertragen und/oder bereitgestellt werden (z. B. an einen Kartenmanager des Servers/der Server 1778 der **Fig. 17D**), und Informationen von dem/den Server(n) können an den Kartenwahrnehmer und/oder einen Lokalisierungsmanager des Fahrzeugs 1700 gesendet, übertragen und/oder bereitgestellt werden. Der Kartenmanager kann eine Cloud-Mapping-Anwendung umfassen, die sich entfernt von dem Fahrzeug 1700 befindet und auf die das Fahrzeug 1700 über ein oder mehrere Netzwerke zugreifen kann. Zum Beispiel kann der Kartenwahrnehmer und/oder der Lokalisierungsmanager des Fahrzeugs 1700 mit dem Kartenmanager und/oder einer oder mehreren anderen Komponenten oder Funktionen des Servers/der Server kommunizieren, um den Kartenwahrnehmer und/oder den Lokalisierungsmanager über vergangene und aktuelle Fahrten oder Fahrten des Fahrzeugs 1700 sowie vergangene und aktuelle Fahrten oder Fahrten anderer Fahrzeuge zu informieren. Der Kartenmanager kann Kartenausgaben (z. B. Kartendaten) bereitstellen, die von dem Lokalisierungsmanager basierend auf eines bestimmten Standorts des Fahrzeugs 1700 lokalisiert werden können, und die lokalisierten Kartenausgaben können von dem Weltmodellmanager 126 verwendet werden, um das Weltmodell zu erzeugen und/oder zu aktualisieren.

[0070] Die Planungskomponente(n) 128 können einen Routenplaner, einen Fahrspurplaner, einen Verhaltensplaner und einen Verhaltensauswähler neben anderen Komponenten, Merkmalen und/oder Funktionen umfassen. Der Routenplaner kann neben anderen Informationen die Informationen des Kartenwahrnehmers, des Kartenmanagers und/oder des Lokalisierungsmanagers verwenden, um einen geplanten Weg zu generieren, der aus GNSS-Wegpunkten (z. B. GPS-Wegpunkten), 3D-Weltkoordinaten (z. B. kartesische, polare usw.), die Koordinaten relativ zu einem Ursprungspunkt auf dem Fahrzeug 1700 angeben, usw. bestehen kann. Die Wegpunkte können für eine bestimmte Entfernung in der Zukunft für das Fahrzeug 1700 repräsentativ sein, wie z. B. eine Anzahl von Stadtblöcken, eine Anzahl von Kilometern, eine Anzahl von Fuß, eine Anzahl von Zoll, eine Anzahl von Meilen usw., die als Ziel für den Fahrspurplaner verwendet werden können.

[0071] Der Fahrspurplaner kann den Fahrspurgraphen (z. B. den Fahrspurgraphen des Wegwahrnehmers), Objektpositionen innerhalb des Fahrspurgraphen (z. B. gemäß dem Lokalisierungsmanager) und/oder einen Zielpunkt und eine Zielrichtung in der Entfernung in der Zukunft vom Routenplaner als Eingaben verwenden. Der Zielpunkt und die Zielrichtung können auf den am besten übereinstimmenden fahrbaren Punkt und die Richtung im Fahrspurgraphen abgebildet werden (z. B. basierend auf GNSS und/oder Kompassrichtung). Ein Graphensuchalgorithmus kann dann auf dem Fahrspurgraphen von einer aktuellen Kante im Fahrspurgraphen aus aus-

geführt werden, um den kürzesten Weg zum Zielpunkt zu finden.

[0072] Der Verhaltensplaner kann die Durchfiihrbarkeit grundlegender Verhaltensweisen des Fahrzeugs 1700 bestimmen, wie z. B. das Beibehalten der Fahrspur oder das Wechseln der Fahrspur nach links oder rechts, so dass die durchführbaren Verhaltensweisen mit den vom Fahrspurplaner ausgegebenen, am meisten gewünschten Verhaltensweisen ausgerichtet werden können. Wenn beispielsweise festgestellt wird, dass das gewünschte Verhalten nicht sicher und/oder nicht verfügbar ist, kann stattdessen ein Standardverhalten ausgewählt werden (z. B. kann das Standardverhalten sein, in der Spur zu bleiben, wenn das gewünschte Verhalten oder der Spurwechsel nicht sicher ist).

[0073] Die Steuerkomponente(n) 130 können einer Trajektorie oder einem Weg (lateral und longitudinal) folgen, die/der von der Verhaltensauswahl (z. B. basierend auf der dichten Erkennungsdaten 120) der Planungskomponente(n) 128 so genau wie möglich und innerhalb der Fähigkeiten des Fahrzeugs 1700 empfangen wurde. Die Steuerkomponente(n) 130 können eine enge Rückkopplung verwenden, um ungeplante Ereignisse oder Verhaltensweisen, die nicht modelliert sind, und/oder alles, was Abweichungen vom Ideal verursacht (z. B. unerwartete Verzögerung), zu behandeln. In einigen Beispielen können die Steuerkomponente(n) 130 ein Vorwärtsprognosemodell verwenden, das die Steuerung als Eingangsvariable verwendet und Vorhersagen erzeugt, die mit dem gewünschten Zustand verglichen werden können (z. B. mit dem gewünschten lateralen und longitudinalen Weg, der von der/den Planungskomponente(n) 128 angefordert wurde). Die Steuerung(en), die die Diskrepanz minimieren, können bestimmt werden.

[0074] Obwohl die Planungskomponente(n) 128 und die Steuerkomponente(n) 130 getrennt dargestellt sind, ist dies nicht als Einschränkung zu verstehen. Zum Beispiel kann in einigen Ausführungsformen die Abgrenzung zwischen der/den Planungskomponente(n) 128 und der/den Steuerkomponente(n) 130 nicht genau definiert sein. So können mindestens einige der Komponenten, Merkmale und/oder Funktionen, die der/den Planungskomponente(n) 128 zugeordnet sind, auch mit der/den Steuerkomponente(n) 130 verbunden sein und umgekehrt. Dies kann auch für jede der separat dargestellten Komponenten des Fahrstapels 122 zutreffen.

[0075] Die Hindernisvermeidungskomponente(n) 132 können das autonome Fahrzeug 1700 bei der Vermeidung von Kollisionen mit Objekten (z. B. beweglichen und stationären Objekten) unterstützen. Die Hindernisvermeidungskomponente(n) 132 können einen Berechnungsmechanismus auf einer „Primärebene“ der Hindernisvermeidung umfassen und als „Überlebensgehim“ oder „Reptiliengehim“ für das Fahrzeug 1700 fungieren. In einigen Beispielen kann die Hindernisvermeidungskomponente(n) 132 unabhängig von Komponenten, Merkmalen und/oder Funktionen des Fahrzeugs 1700 verwendet werden, die erforderlich sind, um Verkehrsregeln zu befolgen und rücksichtsvoll zu fahren. In solchen Beispielen können die Hindernisvermeidungskomponente(n) Verkehrsgesetze, Straßenverkehrsregeln und Normen für rücksichtsvolles Fahren ignorieren, um sicherzustellen, dass es nicht zu Kollisionen zwischen dem Fahrzeug 1700 und Objekten kommt. Daher kann die Schicht zur Hindernisvermeidung eine von der Schicht für die Verkehrsregeln getrennte Schicht sein, und die Schicht zur Hindernisvermeidung kann sicherstellen, dass das Fahrzeug 1700 nur sichere Aktionen unter dem Gesichtspunkt der Hindernisvermeidung durchführt. Die Ebene der Straßenverkehrsregeln kann andererseits sicherstellen, dass das Fahrzeug die Verkehrsgesetze und -konventionen beachtet und die gesetzliche und konventionelle Vorfahrt einhält (wie hier beschrieben).

[0076] In einigen Beispielen können die befahrbaren oder anderen navigierbaren Wege und/oder die dichten Erkennungsdaten 120 von der (den) Hindernisvermeidungskomponente(n) 132 bei der Bestimmung von Steuerungen oder zu ergreifenden Maßnahmen verwendet werden. Beispielsweise können die befahrbaren Wege den Hindernisvermeidungskomponenten 132 einen Hinweis darauf geben, wo das Fahrzeug 1700 manövrieren kann, ohne auf irgendwelche Objekte, Vorsprünge, Strukturen und/oder Ähnliches zu stoßen, oder wo mindestens keine statischen Strukturen vorhanden sein können.

[0077] In nicht einschränkenden Ausführungsformen kann die Hindernisvermeidungskomponente(n) 132 als separates, eigenständiges Merkmal des Fahrzeugs 1700 implementiert werden. Beispielsweise können die Hindernisvermeidungskomponente(n) 132 separat (z. B. parallel zu, vor und/oder nach) der Planungsebene, der Steuerungsebene, der Betätigungsebene und/oder anderen Ebenen des Fahrstapels 122 arbeiten.

[0078] Von daher kann das Fahrzeug 1700 diese Informationen (z. B. als Kanten oder Schienen der Wege) verwenden, um zu navigieren, zu planen oder anderweitig einen oder mehrere Vorgänge (z. B. Spurhaltung, Spurwechsel, Zusammenführen, Aufteilen usw.) innerhalb der Umgebung durchzuführen.

[0079] Bezugnehmend auf die **Fig. 7-9**, umfasst jeder Block der hier beschriebenen Verfahren 700-900 einen Rechenprozess, der mit einer beliebi-

gen Kombination aus Hardware, Firmware und/oder Software durchgeführt werden kann. Beispielsweise können verschiedene Funktionen von einem Prozessor ausgeführt werden, der im Speicher gespeicherte Anweisungen ausführt. Die Verfahren können auch in Form von computerverwendbaren Anweisungen, die auf Computerspeichermedien gespeichert sind, verkörpert sein. Die Methoden können durch eine eigenständige Anwendung, einen Dienst oder einen gehosteten Dienst (eigenständig oder in Kombination mit einem anderen gehosteten Dienst) oder ein Plug-in für ein anderes Produkt bereitgestellt werden, um nur einige zu nennen. Darüber hinaus werden die Methoden 700-900 beispielhaft in Bezug auf die Oberflächenrekonstruktionspipeline 100 der **Fig. 1** beschrieben. Diese Methoden können jedoch zusätzlich oder alternativ von einem beliebigen System oder einer beliebigen Kombination von Systemen ausgeführt werden, einschließlich, aber nicht beschränkt auf die hier beschriebenen Systeme.

[0080] Fig. 7 ist ein Flussdiagramm, das ein Verfahren 700 zur Erzeugung einer Darstellung einer dreidimensionalen (3D) Oberflächenstruktur während einer Erfassungssitzung gemäß einigen Ausführungsformen der vorliegenden Offenbarung zeigt. Das Verfahren 700 umfasst im Block B702 das Erzeugen einer ersten Darstellung einer dreidimensionalen (3D) Oberflächenstruktur einer Komponente der Umgebung mindestens teilweise basierend auf Bilddaten, die während einer Erfassungssitzung mit einer oder mehreren Kameras eines Ego-Objekts in einer Umgebung erzeugt wurden. In Bezug auf **Fig. 1** können beispielsweise eine oder mehrere Kameras 101 eines Ego-Objekts verwendet werden, um die Bilddaten 102 zu erfassen, während das Ego-Objekt durch die Umgebung navigiert, und der 3D-Strukturschätzer 105 kann die Bilddaten 102 verarbeiten, um die 3D-Struktur einer bestimmten Komponente der Umgebung zu schätzen, wie z. B. eine 3D-Straßenoberfläche oder einen anderen Umgebungsteil. Jede geeignete 3D-Strukturschätzungsmethode kann verwendet werden, wie z. B. Structure from Motion (SfM), Stereovision und/oder eine andere 3D-Oberflächenstrukturschätzungsmethode. In einigen Ausführungsformen können eine Segmentierungsmaske oder andere Klassifizierungsdaten verwendet werden, um Punkte aus der geschätzten 3D-Struktur auszuwählen, die sich auf der Komponente der Umgebung von Interesse befinden. Die resultierende Darstellung der 3D-Struktur kann eine 3D-Punktwolke, ein Projektionsbild oder eine andere Darstellung umfassen.

[0081] Das Verfahren 700 umfasst im Block B704 das Erzeugen einer zweiten Darstellung der 3D-Oberflächenstruktur, mindestens teilweise basierend auf dem Verdichten der ersten Darstellung der 3D-Oberflächenstruktur. In Bezug auf **Fig. 1** kann der Erkennungsverdichter 115 beispielsweise die spärlichen Erkennungsdaten 110 verdichten, um eine dichtere Darstellung der 3D-Oberflächenstruktur (z. B. dichte Erkennungsdaten 120) zu erzeugen. Im Allgemeinen können die spärlichen Erkennungsdaten 110 jede geeignete Form annehmen, wie z. B. eine spärliche 3D-Punktwolke oder ein Projektionsbild der spärlichen 3D-Punktwolke (z. B. eine 2D-Höhenkarte von oben nach unten). In einigen Ausführungsformen kann der Erkennungsverdichter 115 die spärlichen Erkennungsdaten 110 unter Verwendung eines oder mehrerer maschineller Lemmodelle verdichten, wie z. B. eines Markov-Zufallsfeldes (z. B. über den Markov-Zufallsfeld-Oberflächenschätzer 310 der **Fig. 3**) und/oder eines oder mehrerer tiefer neuronaler Netze (DNNs) (z. B. über den Deep-Learning-Oberflächenschätzer 320 der **Fig. 3**). Die resultierende Darstellung der 3D-Oberflächenstruktur (z. B. dichte Erkennungsdaten 120) kann jede geeignete Form annehmen, z. B. eine 2D-Höhenkarte und/oder eine 3D-Punktwolke.

[0082] Das Verfahren 700 umfasst im Block B706 die Bereitstellung der zweiten Darstellung der 3D-Oberflächenstruktur an eine Steuerkomponente des Ego-Objekts während der Erfassungssitzung. Beispielsweise können die dichten Erkennungsdaten 120 der **Fig. 1** oder eine andere Darstellung der 3D-Oberflächenstruktur einer oder mehreren Steuerkomponenten des Ego-Objekts (z. B. dem Softwarestapel 122 der **Fig. 1**, Komponenten des autonomen Fahrzeugs 1700 der **Fig. 17A-17D**, wie z. B. Controller 1736, ADAS-System 1738 und/oder SOC(s) 1704) zur Verfügung gestellt und verwendet werden, um das Ego-Objekt bei der Durchführung einer oder mehrerer Operationen innerhalb der Umgebung zu unterstützen, wie z. B. Wegplanung, Vermeidung von Hindernissen oder Höckern, Anpassung eines Aufhängungssystems des Ego-Objekts oder Ego-Actors an die aktuelle Straßenoberfläche, Anwendung einer frühen Beschleunigung oder Verzögerung basierend auf einer sich nähernden Oberflächenneigung, Kartierung und/oder andere.

[0083] Fig. 8 ist ein Flussdiagramm, das ein Verfahren 800 zum Erzeugen einer verdichteten Darstellung einer 3D-Oberflächenstruktur zeigt, die mindestens auf einem Markov-Zufallsfeld basiert, gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Das Verfahren 800 umfasst im Block B802 die Erzeugung einer ersten Darstellung einer dreidimensionalen (3D) Oberflächenstruktur einer Komponente der Umgebung unter Verwendung von Bilddaten von einer oder mehreren Kameras eines Ego-Objekts in einer Umgebung. In Bezug auf **Fig. 1** können beispielsweise eine oder mehrere Kameras 101 eines Ego-Objekts verwendet werden, um die Bilddaten 102 zu erfassen, während sich das Ego-Objekt durch die Umgebung bewegt, und der 3D-Strukturschätzer 105 kann die Bilddaten 102 verarbeiten,

um die 3D-Struktur einer bestimmten interessierenden Oberfläche zu schätzen.

[0084] Das Verfahren 800 umfasst in Block B804 das Erzeugen einer verdichteten Darstellung der 3D-Oberflächenstruktur mindestens basierend auf einem Markov-Zufallsfeld, das eine Beziehung zwischen der ersten Darstellung und der verdichteten Darstellung modelliert. Beispielsweise kann der Markov-Zufallsfeld-Oberflächenschätzer 310 der **Fig. 3** eine Maximum-a-Posterior (MAP)-Inferenz durchführen, um die wahrscheinlichste verdichtete Darstellung (z. B. die dichten Erfassungsdaten 120) zu schätzen, wenn das Markov-Zufallsfeld und die erste Darstellung (z. B. die spärlichen Erfassungsdaten 110) gegeben sind.

[0085] Das Verfahren 800 umfasst in Block B806 die Liefern der verdichteten Darstellung der 3D-Oberflächenstruktur an eine Steuerkomponente des Ego-Objekts. Beispielsweise können die dichten Erkennungsdaten 120 der **Fig. 1** oder eine andere Darstellung der 3D-Oberflächenstruktur einer oder mehreren Steuerkomponenten des Ego-Objekts (z. B. dem Software-Stapel 122 der **Fig. 1**, Komponenten des autonomen Fahrzeugs 1700 der **Fig. 17A-17D**, wie z. B. Controller 1736, ADAS-System 1738 und/oder SOC(s) 1704) bereitgestellt und verwendet werden, um das Ego-Objekt bei der Durchführung einer oder mehrerer Operationen in der Umgebung zu unterstützen.

[0086] **Fig. 9** ist ein Flussdiagramm, das ein Verfahren 900 zum Steuern eines Fahrzeugs zeigt, mindestens teilweise basierend auf einer 3D-Straßenoberflächenstruktur, die unter Verwendung eines oder mehrerer neuronaler Netze geschätzt wird, gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Das Verfahren 900 umfasst im Block B902 den Empfang von Bilddaten, die mit einer oder mehreren Kameras eines Fahrzeugs während des Betriebs des Fahrzeugs in einer Umgebung erzeugt werden. In Bezug auf **Fig. 1** können zum Beispiel eine oder mehrere Kameras 101 eines Fahrzeugs verwendet werden, um die Bilddaten 102 zu erfassen, während das Fahrzeug durch die Umgebung navigiert.

[0087] Das Verfahren 900 umfasst in Block B904 die virtuelle Rekonstruktion einer Straßenoberfläche in der Umgebung während des Betriebs des Fahrzeugs in der Umgebung, mindestens teilweise basierend auf den Blöcken B906 und B908. Das Verfahren 900 umfasst in Block B906 die Erzeugung einer ersten geschätzten 3D-Oberflächenstruktur der Straßenoberfläche unter Verwendung der Bilddaten. In Bezug auf **Fig. 1** kann der 3D-Strukturschätzer 105 zum Beispiel die Bilddaten 102 verarbeiten, um die 3D-Struktur einer bestimmten interessierenden Oberfläche zu schätzen. Das Verfahren 900 umfasst im Block B908 das Erzeugen einer verdichteten geschätzten 3D-Oberflächenstruktur der Straßenoberfläche, mindestens teilweise basierend auf Anwenden der ersten geschätzten 3D-Oberflächenstruktur auf ein oder mehrere neuronale Netze (NNs). Zum Beispiel kann der Deep-Learning-Modell-Oberflächenschätzer 320 der **Fig. 3** oder **Fig. 5** die spärlichen Erkennungsdaten 110 (z. B. eine spärliche 2D-Top-Down-Höhenkarte) verdichten, indem er Werte der dichten Erkennungsdaten 120 (z. B. eine dichte 2D-Top-Down-Höhenkarte) aus den spärlichen Erkennungsdaten 110 unter Verwendung eines oder mehrerer NNs, wie eines oder mehrerer DNNs, ableitet. Als nicht einschränkendes Beispiel kann in einigen Ausführungsformen, in denen die spärlichen Erfassungsdaten 110 eine spärliche 3D-Punktwolke enthalten, der Encoder 515 der **Fig. 5** die spärliche 3D-Punktwolke projizieren, um ein spärliches Projektionsbild (z. B. eine Top-Down-Höhenkarte) zu bilden, und das spärliche Projektionsbild kann in das/die Deep-Learning-Modell(e) 535 eingespeist werden, um die Regressionsdaten 570 (z. B. ein dichtes Projektionsbild wie eine Top-Down-Höhenkarte) und/oder die Konfidenzdaten 580 vorherzusagen.

[0088] Das Verfahren 900 umfasst im Block B910 das Steuern des Fahrzeugs mindestens teilweise basierend auf Daten, die die verdichtete geschätzte 3D-Oberflächenstruktur repräsentieren. Beispielweise können die dichten Erkennungsdaten 120 der **Fig. 1** oder eine andere Darstellung der verdichteten geschätzten 3D-Oberflächenstruktur an die Steuerkomponente(n) des Ego-Objekts (z. B. Software-Stapel 122 der **Fig. 1**, Komponenten des autonomen Fahrzeugs 1700 der **Fig. 17A-17D**, wie z. B. Controller 1736, ADAS-System 1738 und/oder SOC(s) 1704) bereitgestellt und verwendet werden, um das Ego-Objekt bei der Durchführung einer oder mehrerer Operationen in der Umgebung zu unterstützen.

Erzeugen von Trainingsdaten und Trainieren von Deep-Learning-Modellen eines 3D-Oberflächenrekonstruktionssystems

[0089] Um das Training eines Deep-Learning-Modells für ein 3D-Oberflächenrekonstruktionssystem (z. B. das/die Deep-Learning-Modell(e) 535 der **Fig. 5** oder **Fig. 6**) zu unterstützen, kann ein Trainingsdatensatz (z. B. mit spärlichen Eingabedaten und/oder Ground-Truth-Darstellungen der 3D-Oberflächenstruktur) auf verschiedene Weise erzeugt, zusammengestellt und/oder ausgewählt werden. Im Allgemeinen kann die Art der Trainingsdaten von der Architektur des zu trainierenden Deep-Learning-Modells abhängen. Zum Beispiel können bestimmte Implementierungen Eingabe-Trainingsdaten erfordern, die spärliche Darstellungen der 3D-Oberflächenstruktur (z. B. spärliche Höhenkarten) und/oder Bilddaten aus einer anderen Perspektive (z. B. Bilder

einer perspektivischen Ansicht) umfassen, und Ground-Truth-Trainingsdaten, die dichte Darstellungen der 3D-Oberflächenstruktur (z. B. dichte Höhenkarten) und/oder Segmentierungsmasken (z. B. Identifizierung einer gewünschten Oberfläche wie einer Straße oder eines anderen befahrbaren Raums) umfassen. In einigen Ausführungsformen können reale Daten und/oder virtuelle Daten gesammelt und zur Ableitung von Trainingsdaten verwendet werden. Als nicht einschränkendes Beispiel können Trainingsdaten durch das Rendern von Frames virtueller Sensordaten, Segmentierungsmasken und Tiefenkarten, die parametrische mathematische Modellierung einer 3D-Straßenoberfläche, das Sammeln und Beschriften realer Sensordaten von einem einzelnen LiDAR-Sensor und/oder das Sammeln und Beschriften realer Sensordaten, die von mehreren LiDAR-Sensoren gesammelt wurden, erzeugt werden.

[0090] Erzeugen von Trainingsdaten aus einer simulierten Umgebung. In einigen Ausführungsformen können Trainingsdaten erzeugt werden, indem Frames von virtuellen Sensordaten, Segmentierungsmasken und/oder Tiefenkarten, die eine simulierte Umgebung darstellen, gerendert oder erzeugt werden. Beispielsweise kann eine Simulation ausgeführt werden, um eine virtuelle Welt oder Umgebung (z. B. eine simulierte Umgebung) zu simulieren, und ein virtuelles Fahrzeug oder ein anderes Objekt kann innerhalb der simulierten Umgebung simuliert werden. Das virtuelle Fahrzeug oder Objekt kann eine beliebige Anzahl von Sensoren enthalten (z. B. virtuelle oder simulierte Sensoren), und für die Sensoren können virtuelle Sensordaten simuliert werden. So können Frames virtueller Sensordaten (z. B. virtuelle Bilddaten, die dem Sichtfeld der virtuellen Kamera(s) eines virtuellen Fahrzeugs entsprechen) und entsprechende Segmentierungsmasken und Tiefenkarten basierend auf der simulierten Umgebung erzeugt werden. Die virtuellen Sensordaten können verwendet werden, um Eingabetrainingsdaten zu erzeugen (oder als solche zu verwenden), und die Segmentierungsmasken und/oder Tiefenkarten können verwendet werden, um Ground-Truth-Trainingsdaten zu erzeugen (oder als solche zu verwenden).

[0091] Fig. 10 ist ein Datenflussdiagramm, das ein Beispiel für eine Trainingsdaten-Erzeugungspipeline 1000 darstellt, die eine simulierte Umgebung verwendet, gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Die Trainingsdaten-Erzeugungspipeline 1000 umfasst eine Simulatorkomponente 1010, die eine simulierte Umgebung erzeugen kann, sowie Bilder 1020 von virtuellen Sensordaten, Segmentierungsmaske(n) 1030 und/oder Tiefenkarte(n) 1040, die die simulierte Umgebung darstellen. Ein 3D-Strukturschätzer 1050 (der z. B. dem 3D-Strukturschätzer 105 der **Fig. 1** entsprechen kann) kann eine spärliche Darstellung einer 3D-Struktur einer interessierenden Oberfläche (z. B. eine spärliche Punktwolke, ein Projektionsbild) aus Frames 1020 (z. B. einem gerenderten Bild) erzeugen, und die spärliche Darstellung der 3D-Struktur und/oder die Frames 1020 (z. B. ein gerendertes Bild) können als Eingabetrainingsdaten 1080 verwendet werden. Zum Erzeugen von Ground-Truth-Trainingsdaten 1090 kann ein 3D-Punktwolkengenerator 1060 umgekehrt Entfernungswerte aus der/den Tiefenkarte(n) 1040 in 3D-Weltkoordinaten projizieren, indem er die bekannte Position und Orientierung der virtuellen Kamera verwendet, relativ zu der die Entfernungswerte der Tiefenkarte(n) 1040 erzeugt wurden, und der 3D-Punktwolkengenerator 1060 kann die Segmentierungsmaske(n) 1030 verwenden, um 3D-Punkte auf der interessierenden Oberfläche (z. B. einer Straßenoberfläche) herauszufiltern. Da die resultierende 3D-Punktwolke spärlich sein kann, kann ein Nachprozessor 1070 verwendet werden, um fehlende Werte zu interpolieren und eine dichte Darstellung der 3D-Struktur der interessierenden Oberfläche (z. B. eine Punktwolke, ein Projektionsbild) zu erzeugen, und die dichte Darstellung der 3D-Struktur und/oder die Segmentierungsmaske(n) 1030 können als Ground-Truth-Trainingsdaten 1090 verwendet werden.

[0092] Die Simulatorkomponente 1010 kann ein Simulationssystem umfassen, das eine virtuelle Welt oder Umgebung simuliert (z. B. eine simulierte Umgebung). Beispielsweise kann das Simulationssystem eine globale Simulation erzeugen, die eine simulierte Umgebung erzeugt, die Fahrzeuge mit künstlicher Intelligenz (KI) oder andere Objekte (z. B. Fußgänger, Tiere usw.), Hardware-in-the-Loop-Fahrzeuge (HIL) oder andere Objekte, Software-in-the-Loop-Fahrzeuge (SIL) oder andere Objekte und/oder Person-in-the-Loop-Fahrzeuge (PIL) oder andere Objekte enthalten kann. Die simulierte Umgebung kann durch Rasterung, Raytracing, DNNs wie Generative Adversarial Networks (GANs), eine andere Rendering-Technik und/oder eine Kombination davon erzeugt werden. Die simulierte Umgebung kann Merkmale einer Fahrumgebung enthalten, wie z. B. Straßen, Brücken, Tunnel, Straßenschilder, Ampeln, Zebrastreifen, Gebäude, Bäume und Laub, die Sonne, den Mond, Reflexionen, Schatten usw., um eine reale Umgebung zu simulieren. Die globale Simulation kann innerhalb einer Engine (z. B. einer Spiel-Engine) oder einer anderen Software-Entwicklungsumgebung aufrechterhalten werden, die eine Rendering-Engine (z. B. für 2D- und/oder 3D-Grafiken), eine Physik-Engine (z. B. für Kollisionserkennung, Kollisionsreaktion usw.), Sound, Skripting, Animation, künstliche Intelligenz, Networking, Streaming, Speichermanagement, Threading, Lokalisierungsunterstützung, Szenegraphen, Cinematics und/oder andere Funktionen umfassen kann. Ein beispielhaftes Simulationssys-

tem und eine beispielhafte globale Simulation sind in der am 13. März 2020 eingereichten nicht provisorischen US-Patentanmeldung Nr. 16/818,551 mit dem Titel „Sensor Simulation and Learning Sensor Models with Generative Machine Learning Methods“ beschrieben, deren Inhalt hier durch Bezugnahme in vollem Umfang aufgenommen wird.

[0093] In einigen Ausführungsformen kann die Simulatorkomponente 1010 Frames 1020 virtueller Sensordaten (z. B. Bilddaten), Segmentierungsmaske(n) 1030 und/oder Tiefenkarte(n) 1040 erzeugen, die die simulierte Umgebung darstellen. Zum Beispiel kann die Simulatorkomponente 1010 Bilder der simulierten Umgebung aus der Perspektive einer virtuellen Kamera wiedergeben, die auf einem virtuellen Fahrzeug oder einem anderen Objekt in der simulierten Umgebung angeordnet ist. In einigen Ausführungsformen kann die Simulatorkomponente 1010 bekannte Koordinaten einer simulierten interessierenden Oberfläche (z. B. einer Straßenoberfläche) in der simulierten Umgebung verwenden, um Segmentierungsmaske(n) 1030 und/oder Tiefenkarte(n) 1040 (z. B. Tiefenkarte(n) pro Pixel) zu erzeugen, die dem Bild bzw. den Bildern 1020 der virtuellen Sensordaten entsprechen. Die Frames 1020 der virtuellen Sensordaten, die Segmentierungsmaske(n) 1030 und/oder die Tiefenkarte(n) 1040 (zusammen simulierte oder virtuelle Daten) können gruppiert werden, und die Simulatorkomponente 1010 kann simulierte oder virtuelle Daten erzeugen, die aufeinanderfolgende Zeitabschnitte in der simulierten Umgebung darstellen, z. B. wenn das virtuelle Fahrzeug oder ein anderes Objekt durch die simulierte Umgebung navigiert. So kann die Simulatorkomponente 1010 Frames 1020 virtueller Sensordaten, Segmentierungsmaske(n) 1030 und/oder Tiefenkarte(n) 1040 erzeugen, die realistische (z. B. Fahr-)Szenarien darstellen.

[0094] Für jeden gegebenen Frame 1020 kann der 3D-Strukturschätzer 1050 eine 3D-Oberflächenstruktur einer interessierenden Oberfläche (z. B. einer Straßenoberfläche) aus dem Frame 1020 schätzen. Zum Beispiel kann die 3D-Struktur mit Hilfe der hierin beschriebenen Techniken in Bezug auf den 3D-Strukturschätzer 105 der **Fig. 1** geschätzt werden (z. B. unter Verwendung von Structure from Motion, Stereovision, Ausreißerentfernung und/oder Oberflächenpunktauswahl). In einigen Ausführungsformen kann der 3D-Strukturschätzer 1050 die Segmentierungsmaske(n) 1030 verwenden, um Punkte aus einer geschätzten 3D-Struktur auszuwählen, die zu einer Klasse gehören, die durch die Segmentierungsmaske repräsentiert wird (z. B. Punkte, die zu einer interessierenden Oberfläche gehören, wie eine 3D-Straßenoberfläche). In einigen Ausführungsformen können die resultierenden Punkte projiziert werden, um ein Projektionsbild (z. B. eine 2D-Höhenkarte) zu erstellen. Das Ergebnis kann eine spärliche Darstellung der 3D-Struktur der interessierenden Oberfläche sein (z. B. eine spärliche Punktwolke, ein spärliches Projektionsbild). Die spärliche Darstellung der 3D-Struktur und/oder die Frames 1020 (z. B. ein gerendertes Bild) können als Eingabetrainingsdaten 1080 bezeichnet und in einen Trainingsdatensatz aufgenommen werden.

[0095] In einigen Ausführungsformen kann der 3D-Punktwolken-Generator 1060 eine 3D-Punktwolke oder eine andere Darstellung der 3D-Struktur unter Verwendung der Tiefenkarte(n) 1040 erzeugen, um entsprechende Ground-Truth-Trainingsdaten 1090 zu erzeugen. Beispielsweise kann der 3D-Punktwolkengenerator 1060 3D-Punkte erzeugen, indem er Entfernungswerte aus der/den Tiefenkarte(n) 1040 umgekehrt in 3D-Weltkoordinaten der simulierten Umgebung projiziert, wobei er den Standort und die Ausrichtung der virtuellen Kamera verwendet, relativ zu der die Entfernungswerte der Tiefenkarte(n) 1040 erzeugt wurden, und der 3D-Punktwolkengenerator 1060 kann 3D-Punkte auf der interessierenden Oberfläche unter Verwendung der Segmentierungsmaske(n) 1030 auswählen (z. B. durch Auswählen von 3D-Punkten, die auf einen Teil der Segmentierungsmaske(n) 1030 projiziert werden, der die interessierende Oberfläche darstellt). Zusätzlich oder alternativ kann der 3D-Punktwolkengenerator 1060 die Segmentierungsmaske(n) 1030 verwenden, um Entfernungswerte aus der/den Tiefenkarte(n) 1040 für Punkte auszuwählen, die sich auf der interessierenden Oberfläche befinden (z. B. durch Überlagern der Segmentierungsmaske(n) 1030 auf der/den Tiefenkarte(n) 1040), und der 3D-Punktwolkengenerator 1060 kann die ausgewählten Entfernungswerte umgekehrt in die simulierte Umgebung projizieren, um die 3D-Punkte auf der interessierenden Oberfläche zu erzeugen.

[0096] Da die resultierenden 3D-Punkte (z. B. eine 3D-Punktwolke) spärlich sein können, kann der Nachprozessor 1070 verwendet werden, um fehlende Werte mithilfe eines Triangulationsalgorithmus zu interpolieren. Zum Beispiel kann der Nachprozessor 1070 eine Delaunay-Triangulierung in 2D und/oder 3D durchführen. In einem Ausführungsbeispiel mit 2D-Triangulation kann der Nachprozessor 1070 die 3D-Punkte auf die interessierende Oberfläche projizieren, um ein Projektionsbild (z. B. eine 2D-Höhenkarte) zu bilden, und eine Delaunay-Triangulierung in dem Projektionsbild durchführen, um Dreiecke zu erzeugen, und der Nachprozessor 1070 kann Punkte aus den Dreiecken abtasten, um eine gewünschte Anzahl von Punkten für ein dichtes Ground-Truth-Projektionsbild (z. B. eine 2D-Ground-Truth-Höhenkarte) zu erzeugen. In einem Ausführungsbeispiel mit 3D-Triangulation kann der Nachprozessor 1070 eine 3D-Delaunay-Triangulierung durchführen, um ein Oberflächengitter aus Dreiecken zu berechnen, das die 3D-Punkte auf der inte-

ressierenden Oberfläche umgibt, und 3D-Punkte aus den Dreiecken des Oberflächengitters abtasten, um eine gewünschte Anzahl von Punkten für ein dichtes Ground-Truth-Projektionsbild (z. B. eine 2D-Ground-Truth-Höhenkarte) zu erzeugen. Beispielsweise kann der Nachprozessor 1070 3D-Punkte aus dem Oberflächengitter abtasten und die abgetasteten 3D-Punkte projizieren, um ein Ground-Truth-Projektionsbild (z. B. eine Ground-Truth-2D-Höhenkarte) zu erstellen. Pixel in einem Ground-Truth-Projektionsbild, die keine abgetasteten Punkte darstellen, können auf null gesetzt werden. So kann ein dichtes Projektionsbild oder eine andere Darstellung der 3D-Punkte auf der interessierenden Oberfläche und/oder der Segmentierungsmaske(n) 1030 als Ground-Truth-Trainingsdaten 1090 bezeichnet, mit entsprechenden Eingabetrainingsdaten 1090 gepaart und in einen Trainingsdatensatz aufgenommen werden.

[0097] Erzeugen synthetischer Trainingsdaten unter Verwendung parametrischer Modellierung. In einem anderen beispielhaften Verfahren zum Erzeugen von Trainingsdaten können synthetische Trainingsdaten durch parametrische mathematische Modellierung einer gewünschten Oberfläche, wie z. B. einer 3D-Straßenoberfläche, erzeugt werden. Beispielsweise kann eine Vielzahl von synthetischen 3D-Straßenoberflächen durch Modellierung einer 3D-Straßenoberfläche mit unterschiedlichen Parametern erzeugt werden, um Änderungen der Straßenrichtung und der lateralen Oberflächenneigung zu simulieren. Als nicht einschränkendes Beispiel kann eine synthetische 3D-Oberfläche durch Modellierung einer 3D-Kurve auf der synthetischen 3D-Oberfläche und Erweiterung der 3D-Kurve zu einer 3D-Oberfläche erstellt werden. Die sich daraus ergebende synthetische 3D-Oberfläche (oder ihre Kurvenkomponenten) kann abgetastet werden, und die abgetasteten Punkte können projiziert werden, um ein synthetisches Ground-Truth-Projektionsbild zu erstellen (z. B. eine 2D-Höhenkarte). Um entsprechende Eingabetrainingsdaten zu erzeugen, kann ein bekanntes Muster, das darstellt, welche Pixel während der 3D-Strukturschätzung unbeobachtet bleiben können, erzeugt und auf ein Ground-Truth-Projektionsbild angewendet werden, um ein entsprechendes spärliches Projektionsbild mit unbeobachteten Werten zu simulieren. Auf diese Weise können synthetische spärliche Eingangsprojektionsbilder und dichte Ground-Truth-Projektionsbilder erzeugt und in einen Trainingsdatensatz aufgenommen werden.

[0098] **Fig. 11** ist eine Darstellung eines beispielhaften parametrischen mathematischen Modells einer gewünschten Oberfläche gemäß einigen Ausführungsformen der vorliegenden Offenbarung. In dem in **Fig. 11** dargestellten Beispiel wird eine 3D-Oberfläche mit der longitudinalen Kurve l und den lateralen Kurven q_j modelliert. In einem Ausführungsbeispiel, in dem die modellierte 3D-Oberfläche eine 3D-Straßenoberfläche ist, können die Parameter der parametrischen Gleichungen, die die longitudinale Kurve l und die lateralen Kurven q_j definieren, variiert werden, um verschiedene Arten von 3D-Straßenoberflächen zu simulieren.

[0099] Als nicht einschränkendes Beispiel kann eine 3D-Kurve auf einer synthetischen 3D-Straßenoberfläche durch Abtasten von longitudinalen, lateralen und Höhenwerten für die 3D-Kurve erzeugt werden. Zum Beispiel kann ein gewünschter Satz von longitudinalen Werten $[x_0,...,x_n]$ für eine synthetische 3D-Kurve auf einer synthetischen 3D-Oberfläche anfänglich abgetastet oder anderweitig ausgewählt werden. Für eine beispielhafte Straßenoberfläche können die longitudinalen Werte einen gewünschten Wahrnehmungsbereich für einen Deep-Learning-Modell-Oberflächenschätzer darstellen, z. B. 0 bis 300 m. In einigen Ausführungsformen können die lateralen Werte für die synthetische 3D-Kurve als Polynom zweiter Ordnung der longitudinalen Werte berechnet werden x: $y = a x^2 + bx + c$. In Ausführungsformen mit synthetischen 3D-Straßenoberflächen können mehrere synthetische 3D-Kurven erzeugt werden, indem verschiedene Werte für die Polynomkonstanten a, b und/oder c abgetastet werden, um verschiedene Änderungen der Straßenrichtung (z. B. Kurven, Abbiegungen usw.) für verschiedene synthetische 3D-Kurven zu simulieren. In einigen Ausführungsformen können die Höhenwerte für die synthetische 3D-Kurve als Linearkombination von Fourier-Basen berechnet werden:

$$z = \sum_{k=1}^{K} \quad c[k] * cos\left(f[k] * x\right) \qquad \text{Gl. (5)}$$

wobei K die Anzahl der Fourier-Basen, c eine Gewichtung für eine bestimmte Basis k und f die Frequenz für eine bestimmte Basis k ist. In Ausführungsformen mit synthetischen 3D-Straßenoberflächen können für verschiedene synthetische 3D-Kurven unterschiedliche Höhenwerte berechnet werden, indem unterschiedliche Abtastwerte für die Anzahl der Basen K, die Gewichtung c für eine bestimmte Basis k und/oder die Frequenz f für eine bestimmte Basis k verwendet werden, um unterschiedliche Änderungen der Oberflächenhöhe für verschiedene synthetische 3D-Kurven zu simulieren. Das Ergebnis kann eine longitudinale 3D-Kurve sein, die in dem in **Fig. 11** dargestellten Beispiel durch die Kurve l repräsentiert wird.

[0100] In einigen Ausführungsformen kann die longitudinale 3D-Kurve zu einer 3D-Oberfläche erweitert werden. Zum Beispiel kann eine longitudinale 3D-Kurve eine beliebige Anzahl von Punkten $\{x_j, y_j, z_j\}$ for j in [1,...,n] enthalten und jeder gegebene Punkt auf der longitudinalen 3D-Kurve (z. B. jeder Punkt) kann zu einer entsprechenden lateralen 3D-Kurve

erweitert werden, die in dem in **Fig. 11** dargestellten Beispiel durch die Kurven q_j dargestellt wird. Zum Beispiel kann ein Parameter α definiert werden, um den Winkel zwischen einer synthetischen 3D-Oberfläche (z. B. der synthetischen 3D-Straßenoberfläche) und einer Oberfläche (z. B. der Bodenebene) zu bezeichnen, z = 0), und verschiedene Werte von α können abgetastet werden, um verschiedene laterale Oberflächenneigungen an verschiedenen Punkten der longitudinalen 3D-Kurve und/oder für verschiedene synthetische 3D-Kurven zu simulieren. Für einen bestimmten 3D-Punkt $p_j = \{x_j, y_j, z_j\}$ auf der longitudinalen 3D-Kurve l kann der 3D-Punkt zu einer lateralen 3D-Kurve q_j erweitert werden, die durch p_j senkrecht zur Kurve l bei p_j verläuft und einen Winkel α relativ zur Oberfläche z = 0 aufweist. Jede Art von lateraler 3D-Kurve kann verwendet werden (z. B. linear, polynomisch usw.), und jede gegebene laterale 3D-Kurve q_j kann m-mal abgetastet werden, um einen entsprechenden 3D-Punkt zu erweitern $p_j = \{x_j, y_j, z_j\}$ auf einer longitudinalen 3D-Kurve l in eine Menge von 3D-Punkten $\{x_{ij}, y_{ij}, z_{ij}\}$, i = [1, ... , m], wobei unterschiedliche Werte von m abgetastet werden können, um unterschiedliche Straßenbreiten an unterschiedlichen Punkten der longitudinalen 3D-Kurve und/oder für unterschiedliche synthetische 3D-Oberflächen zu simulieren. Der Prozess kann für jeden gegebenen Punkt auf einer longitudinalen 3D-Kurve (z. B. jeden Punkt) wiederholt werden, um eine dichte 3D-Punktwolke zu erzeugen, die projiziert werden kann, um ein Ground-Truth-Projektionsbild (z. B. eine 2D-Ground-Truth-Höhenkarte) zu bilden.

[0101] Um entsprechende Eingabetrainingsdaten zu erzeugen, kann ein bekanntes Muster, das darstellt, welche Pixel bei der 3D-Schätzung unbeobachtet bleiben können, erzeugt und auf das Ground-Truth-Projektionsbild angewendet werden, um eine Teilmenge von Pixelwerten zu löschen (z. B. indem diese Pixelwerte auf null gesetzt werden), um unbeobachtete Werte zu simulieren. Nehmen wir zum Beispiel an, eine Ground-Truth-Höhenkarte hat die Größe HxW. In diesem Beispiel kann ein Muster, das durch N binäre Karten der Größe HxW dargestellt wird, durch die Durchführung einer 3D-Schätzung an realen Daten erzeugt werden. Beispielsweise können ein oder mehrere Fahrzeuge (z. B. das Fahrzeug 1700 in den **Fig. 17A-D**) Sensordaten (z. B. Bilddaten) von einem oder mehreren Sensoren (z. B. Kameras) des Fahrzeugs/der Fahrzeuge in realen (z. B. physikalischen) Umgebungen erfassen, wie im Folgenden näher erläutert wird. Eine 3D-Oberflächenstruktur einer gewünschten Oberfläche (z. B. 3D-Straßenoberfläche) kann aus jedem Bild der Sensordaten (wie hierin beschrieben) geschätzt werden, und die sich ergebende Darstellung der 3D-Struktur (z. B. eine spärliche 3D-Punktwolke) kann projiziert werden, um ein spärliches Projektionsbild (z. B. eine spärliche 2D-Höhenkarte) zu bilden, das sowohl beobachtete als auch unbeobachtete Werte enthalten kann. Für jedes von N spärlichen Projektionsbildern der Größe HxW kann eine entsprechende binäre Karte der Größe HxW erzeugt werden, um darzustellen, welche Pixel beobachtet und unbeobachtet sind. Beispielsweise können Pixel einer binären Karte, die beobachteten Werten entsprechen, auf 1 gesetzt werden, und Pixel, die unbeobachteten Werten entsprechen, auf 0. Auf diese Weise kann ein NxHxW-Muster von binären Karten erzeugt werden, um darzustellen, welche Pixel bei der 3D-Schätzung unbeobachtet bleiben können.

[0102] Für jedes synthetische Ground-Truth-Projektionsbild kann eine der N binären Karten zufällig abgetastet und auf das synthetische Ground-Truth-Projektionsbild angewandt werden (z. B. durch elementweise Multiplikation), um ein entsprechendes synthetisches spärliches Projektionsbild zu erzeugen. Auf diese Weise können Paare von synthetischen Eingabe- und Ground-Truth-Projektionsbildern erzeugt und zu einem Trainingsdatensatz hinzugefügt werden.

[0103] Erzeugen von Trainingsdaten aus realen Sensordaten. In einigen Ausführungsformen können Trainingsdaten durch das Sammeln und Annotieren von realen Sensordaten erzeugt werden. Zum Beispiel können ein oder mehrere Fahrzeuge Frames von Sensordaten (z. B. Bilddaten und LiDAR-Daten) von einem oder mehreren Sensoren (z. B. Kamera(s) und LiDAR-Sensor(en)) des Fahrzeugs (der Fahrzeuge) in realen (z. B. physikalischen) Umgebungen sammeln. In einigen Ausführungsformen können LiDAR-Daten geglättet, von Ausreißern befreit, einer Triangulation unterzogen werden, um fehlende Werte zu interpolieren, von mehreren LiDAR-Sensoren akkumuliert, zeitlich und/oder räumlich mit entsprechenden Bilddaten ausgerichtet und annotiert werden, um 3D-Punkte auf einer interessierenden Oberfläche (z. B. einer 3D-Straßenoberfläche) zu identifizieren. Eine Darstellung der identifizierten 3D-Punkte (z. B. eine 3D-Punktwolke, ein Projektionsbild) kann als Ground-Truth-Trainingsdaten bezeichnet werden. In einigen Ausführungsformen können die Objekterkennung, die Freiraumschätzung und/oder die Bildsegmentierung auf Frames von Bilddaten angewendet werden, um entsprechende Segmentierungsmasken zu erzeugen, die als Ground-Truth-Trainingsdaten bezeichnet werden können. Entsprechende Frames von Bilddaten können einer 3D-Schätzung unterzogen werden, und die sich daraus ergebende spärliche Darstellung der interessierenden Oberfläche (z. B. eine 3D-Punktwolke, ein Projektionsbild) kann als Eingabetrainingsdaten bezeichnet werden. So können beispielsweise ein entsprechendes spärliches Projektionsbild, ein Kamerabild, ein dichtes Projektionsbild und/oder eine Segmentierungsmaske gruppiert und in einen Trainingsdatensatz aufgenommen werden.

[0104] **Fig. 12** ist ein Datenflussdiagramm, das eine beispielhafte Ground-Truth-Erzeugungspipeline 1200 darstellt, die gesammelte reale Daten verwendet, gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Die beispielhafte Ground-Truth-Erzeugungspipeline 1200 umfasst eine Aufzeichnungs-Engine 1205, einen 3D-Strukturschätzer 1220, einen Freiraumschätzer 1225, einen Vorprozessor 1240, einen Aligner 1250 und eine Annotationskomponente 1260.

[0105] In einigen Ausführungsformen können ein oder mehrere Datensammelfahrzeuge (z. B. das Fahrzeug 1700 in den **Fig. 17A-D**) mit einer oder mehreren Kameras und LiDAR-Sensoren ausgestattet sein, und eine mit jedem Datensammelfahrzeug verbundene Aufzeichnungs-Engine 1205 kann Sensordaten aufzeichnen, während das Fahrzeug durch reale (z. B. physikalische) Umgebungen fährt. Im Allgemeinen kann ein Datensammelfahrzeug mit einer beliebigen Anzahl und einem beliebigen Typ von Sensoren ausgestattet sein (einschließlich, aber nicht beschränkt auf die in den **Fig. 17A-17C** dargestellten Sensoren). Zum Beispiel kann eine Anzahl von Kameras (z. B. Stereokamera(s) 1768, Weitwinkelkamera(s) 1770 (z. B. Fischaugenkameras), Infrarotkamera(s) 1772, Surround-Kamera(s) 1774 (z. B., 360-Grad-Kameras) und/oder Fern- und/oder Mittelbereichskamera(s) 1798), LIDAR-Sensoren 1764 und/oder andere Sensortypen können so am Fahrzeug positioniert werden, dass es zu einer Überlappung zwischen den Sichtfeldern der Kameras und den Sicht- oder Sensorfeldern der Sensoren kommt. Die räumliche Anordnung der Sensoren kann in einigen Ausführungsformen durch Selbstkalibrierungsalgorithmen kalibriert werden, und die Synchronisation der Sensoren kann gesteuert werden, um eine zeitliche Ausrichtung der Sensorerfassungen zu erreichen. Von daher kann die Aufzeichnungs-Engine 1205 Frames 1210 von einer oder mehreren Kameras und/oder LiDAR-Daten 1215 von einem oder mehreren LiDAR-Sensoren erfassen.

[0106] In einigen Ausführungsformen können die LiDAR-Daten 1215 verwendet werden, um Ground-Truth-Trainingsdaten zu erzeugen. In dem in **Fig. 12** dargestellten Beispiel fuhrt der Vorprozessor 1240 eine oder mehrere Verarbeitungsoperationen an den LiDAR-Daten 1215 vor der Kennzeichnung durch. In einigen Ausführungsformen kann der Vorprozessor 1240 beispielsweise eine zeitliche Glättung durchführen, die einen Zustandsschätzer wie einen Kalman-Filter umfassen kann. Die zeitliche Glättung kann im 3D-Weltraum relativ zum Datensammelfahrzeug, im 3D-Weltraum relativ zu einem festen Ursprung im Weltraum oder in einer Vogelperspektive im 2D-Weltraum durchgeführt werden. In einigen Ausführungsformen kann der Vorprozessor 1240 eine Ausreißerentfernung an den LiDAR-Daten 1215 durchführen (z. B. ähnlich der hierin beschriebenen Technik in Bezug auf den Ausreißerentferner 220 der **Fig. 2**). In einigen Fällen können die resultierenden LiDAR-Daten immer noch spärlich sein. In einigen Ausführungsformen kann der Vorprozessor 1240 fehlende Werte mit Hilfe eines Triangulationsalgorithmus interpolieren (z. B. wie hier in Bezug auf den Nachprozessor 1070 der **Fig. 10** beschrieben). Zusätzlich oder alternativ kann der Vorprozessor 1240 LiDAR-Daten von mehreren LiDAR-Sensoren akkumulieren, um die resultierenden LiDAR-Daten zu verdichten. Zur Veranschaulichung zeigt **Fig. 13A** ein Beispiel für LiDAR-Daten, die von einem einzelnen LiDAR-Scan gesammelt wurden, und **Fig. 13B** zeigt ein Beispiel für LiDAR-Daten, die von mehreren LiDAR-Scans (z. B. von mehreren LiDAR-Sensoren) akkumuliert wurden. Dies sind nur einige Beispiele, und andere Arten von Vorverarbeitungsvorgängen können zusätzlich oder alternativ durchgeführt werden.

[0107] In einigen Ausführungsformen kann der Aligner 1250 die LiDAR-Daten zeitlich mit den entsprechenden Frames 1210 von Bilddaten ausrichten. Im Allgemeinen können Sensordaten von verschiedenen Sensoren mit unterschiedlichen Frequenzen aus verschiedenen Gründen erhalten werden, wie z. B. Unterschiede in den Verzögerungsleitungen, Unterschiede in den Abtastfrequenzen (z. B. Kameras, die mit 30 fps laufen, gegenüber LiDAR, das mit 10 fps läuft), unterschiedliche Auslösezeiten und andere Gründe. Um die Gruppierung und/oder Darstellung von Sensordaten ähnlicher Weltzustände (z. B. Sensordaten, die im Wesentlichen zur gleichen Zeit erfasst wurden) zu erleichtern, kann eine zeitliche Abstimmung vorgenommen werden, um die Sensordaten der verschiedenen Sensoren zu synchronisieren. So kann beispielsweise ein bestimmter Sensor als Referenzsensor verwendet werden, und andere Sensoren können als untergeordnete Sensoren bezeichnet werden. Für einen gegebenen Frame von Sensordaten des Referenzsensors (ein Referenzframe) kann ein Versatz, z. B. ein Zeitdelta, zwischen dem Referenzframe und dem zeitlich nächstgelegenen Frame von Sensordaten von jedem untergeordneten Sensor ermittelt werden. Der Versatz für jeden untergeordneten Sensor kann aufgezeichnet und/oder auf die Erfassungszeiten oder einen anderen Index für die Sensordaten des untergeordneten Sensors angewendet werden. So kann die Bestimmung und/oder Anwendung von Offsets pro Sensor dazu dienen, die verschiedenen Arten von Sensordaten zeitlich auszurichten (z. B. durch Angleichung ihrer Indizes). Beispieltechniken für das Ausrichten von Sensordaten von verschiedenen Sensortypen sind in der nicht provisorischen US-Patentanmeldung Nr. 17/187,350 beschrieben, die am 26. April 2021 eingereicht wurde und den Titel „Ground Truth Data Generation for Deep Neural Network Perception in Autonomous Driving Applica-

tions“ trägt, deren Inhalt hier durch Bezugnahme in vollem Umfang aufgenommen wird.

[0108] Zusätzlich oder alternativ kann der Aligner 1250 die LiDAR-Daten mit den entsprechenden Frames 1210 der Bilddaten räumlich ausrichten, um verschiedene Arten von Sensordaten abzugleichen, die das gleiche Objekt oder einen anderen Teil der Umgebung darstellen. Beispielsweise können LIDAR-Datenpunkte mit Pixeln im Bildraum korreliert werden, indem die relative Ausrichtung, der Standort, die Sichtfelder und ähnliches zwischen dem LiDAR-Sensor, der den LiDAR-Datenpunkt erfasst hat, und der Kamera, die die Bilddaten erzeugt hat, verwendet werden. Techniken zur Korrelation von Sensordaten verschiedener Sensoren sind in der vorläufigen US-Patentanmeldung Nr. 62/514,404, die am 15. März 2019 eingereicht wurde und den Titel „Sequential Neural Network-Based Temporal Information Prediction in Autonomous Driving Applications“ trägt, und in der nicht vorläufigen US-Patentanmeldung Nr. 16/514,404, die am 17. Juli 2019 eingereicht wurde und den Titel „Temporal Information Prediction in Autonomous Machine Applications“ trägt, beschrieben, deren Inhalt hierin jeweils in vollem Umfang enthalten ist.

[0109] In einigen Ausführungsformen können die LiDAR-Daten annotiert werden, um Punkte auf einer interessierenden 3D-Oberfläche (z. B. eine 3D-Straßenoberfläche) zu identifizieren. Im Allgemeinen können die Annotationen synthetisch (z. B. aus Computermodellen oder Renderings), real (z. B. aus realen Daten), maschinell (z. B, maschinell (z. B. durch Merkmalsanalyse und Lernen, um Merkmale aus Daten zu extrahieren und dann Kennzeichnungen zu erzeugen), von Menschen erstellt (z. B. von einem Annotationsexperten, der die Annotationen eingibt) und/oder eine Kombination davon (z. B. ein Mensch identifiziert die Eckpunkte von Polylinien, eine Maschine erzeugt Polygone unter Verwendung eines Polygon-Rasterizers).

[0110] In einigen Ausführungsformen kann die Annotationskomponente 1260 ein Software-Tool (auch als Kennzeichnungswerkzeug bezeichnet) wie z. B. ein Web-Tool umfassen. Es kann eine Abfolge von Annotationsszenen (z. B. Sätze von ausgerichteten LiDAR-Daten und Bilddaten, die ungefähr zur gleichen Zeit aufgenommen wurden) erzeugt werden, eine entsprechende Kennzeichnungsaufgabe bzw. entsprechende Kennzeichnungsaufgaben können in dem Kennzeichnungswerkzeug kodiert werden, und die Annotationen können mit dem Softwarewerkzeug erzeugt werden. In einigen Ausführungsformen kann das Kennzeichnungswerkzeug die ausgerichteten LiDAR-Daten und Bilddaten in einer Kennzeichnungsszene für einen menschlichen Kennzeichner darstellen (z. B. nebeneinander) und/oder Informationen können über die verschiedenen Arten von Sensordaten projiziert werden, um nützliche Kontextinformationen bereitzustellen, wie z. B. Korrespondenzen zwischen den verschiedenen Arten von Sensordaten. Das Kennzeichnungswerkzeug kann Eingaben akzeptieren, die Ground-Truth-Annotationen angeben, die Punkte auf einer interessierenden Oberfläche identifizieren (z. B. 3D-Punkte, Grenzen, eingeschlossene Regionen, Klassenkennzeichnungen), und das Kennzeichnungswerkzeug kann die Annotationen mit den Sensordaten verknüpfen. Ein Beispiel für ein Kennzeichnungswerkzeug ist in der nicht provisorischen US-Patentanmeldung Nr. 17/187,350 beschrieben, die am 26. April 2021 eingereicht wurde und den Titel „Ground Truth Data Generation for Deep Neural Network Perception in Autonomous Driving Applications“ trägt. Von daher kann die Annotationskomponente 1260 Eingaben akzeptieren, die Punkte auf einer interessierenden Oberfläche identifizieren (z. B. eine 3D-Punktwolke, ein Projektionsbild), und eine Darstellung der identifizierten Punkte erzeugen, die mit der Ansicht, Größe und Dimensionalität der Ausgabe(n) des/der zu trainierenden Deep-Learning-Modells/Modelle übereinstimmt, was als Ground-Truth-Trainingsdaten 1295 bezeichnet werden kann.

[0111] In einigen Ausführungsformen können Ground-Truth-Segmentierungsmaske(n) aus Frames 1210 der Bilddaten erzeugt werden. Zum Beispiel kann ein Freiraumschätzer 1225 eine Freiraumschätzung und/oder eine Bildsegmentierung an den erfassten Bildern durchführen, um Regionen (z. B. Pixel) der Bilddaten zu klassifizieren, zu segmentieren und/oder vorherzusagen, die Teil einer gewünschten Klasse sind (z. B. eine Straßenoberfläche). Beispielsweise können mehrere maschinelle Lemmodelle (z. B. ein neuronales Faltungsnetzwerk) trainiert werden, um eine oder mehrere Segmentierungsmaske(n) 1230 und/oder Konfidenzkarten vorherzusagen, die Pixel repräsentieren, die zu einer befahrbaren Straßenoberfläche oder einem anderen befahrbaren Raum, anderen Teilen der Umgebung (z. B. Bürgersteige, Gebäude), belebten Objekten und/oder anderen Klassen gehören. Die Segmentierungsmaske(n) 1230 oder eine andere Darstellung einer erkannten Oberfläche kann als Ground-Truth-Trainingsdaten 1295 bezeichnet werden.

[0112] Um entsprechende Eingabetrainingsdaten zu erzeugen, kann der 3D-Strukturschätzer 1220 für ein beliebiges gegebenes Frame bzw. beliebige gegebene Frames 1210 von Bilddaten eine 3D-Oberflächenstruktur einer interessierenden Oberfläche (z. B. eine Straßenoberfläche) aus dem Frame bzw. den Frames 1210 schätzen (z. B. wie oben in Bezug auf den 3D-Strukturschätzer 105 der **Fig. 1** und/oder den 3D-Strukturschätzer 1050 der **Fig. 10** beschrieben). In einigen Ausführungsformen kann der 3D-Strukturschätzer 1220 die Segmentierungsmaske(n) 1230 verwenden, um Punkte aus einer

geschätzten 3D-Struktur auszuwählen, die zu einer Klasse gehören, die durch die Segmentierungsmaske repräsentiert wird (z. B. Punkte, die zu einer interessierenden Oberfläche gehören, wie eine 3D-Straßenoberfläche). In einigen Ausführungsformen können die resultierenden Punkte projiziert werden, um ein Projektionsbild (z. B. eine 2D-Höhenkarte) zu erstellen. Das Ergebnis kann eine spärliche Darstellung der 3D-Struktur der interessierenden Oberfläche sein (z. B. eine spärliche Punktwolke, ein spärliches Projektionsbild). Die spärliche Darstellung der 3D-Struktur und/oder die Frames 1210 der Bilddaten können als Eingabetrainingsdaten 1290 bezeichnet werden.

[0113] Von daher können die Eingabe-Trainingsdaten 1290 mit entsprechenden Ground-Truth-Trainingsdaten 1295 (z. B. einem dichten Projektionsbild oder einer anderen Darstellung der 3D-Punkte auf der interessierenden Oberfläche und/oder der Segmentierungsmaske(n) 1230) gepaart und in einen Trainingsdatensatz aufgenommen werden.

[0114] Training. In einigen Ausführungsformen kann ein Trainingsdatensatz für ein Deep-Learning-Modell bzw. für Deep-Learning-Modelle für ein 3D-Oberflächenrekonstruktionssystem (z. B. das Deep-Learning-Modell bzw. die Deep-Learning-Modelle 535 der **Fig. 5** oder **Fig. 6**) kann basierend auf der Eingaben und Ausgaben des zu trainierenden Deep-Learning-Modells erzeugt, zusammengestellt und/oder ausgewählt werden. Beispielsweise können bestimmte Implementierungen ein Eingabetraining erfordern, das spärliche Darstellungen der 3D-Oberflächenstruktur (z. B. spärliche Höhenkarten) und/oder Bilddaten aus einer anderen Perspektive (z. B. Bilder einer perspektivischen Ansicht) umfasst, und Ground-Truth-Trainingsdaten, die dichte Darstellungen der 3D-Oberflächenstruktur (z. B. dichte Höhenkarten) und/oder Segmentierungsmasken (z. B. Identifizierung einer gewünschten Oberfläche wie einer Straße oder eines anderen befahrbaren Raums) umfassen. Von daher kann ein Trainingsdatensatz mit Eingabe- und Ground-Truth-Daten, die der Ansicht, Größe und Dimensionalität der Eingabe(n) und Ausgabe(n) eines gewünschten Deep-Learning-Modells entsprechen, mit den hier beschriebenen Techniken erhalten werden, und das/die Deep-Learning-Modell(e) können mit dem ausgewählten Trainingsdatensatz trainiert werden. In Ausführungsformen, in denen das Deep-Learning-Modell eine oder mehrere rekurrente Schichten (z. B. Gated Recurrent Units, Long Short Term Memory) enthält, können die Eingabetrainingsdaten mehrere Frames (z. B. aus aufeinanderfolgenden Zeitscheiben) als ein einziges Sample enthalten.

[0115] Im Allgemeinen kann jede geeignete Verlustfunktion verwendet werden, um das/die Deep-Learning-Modell(e) während des Trainings zu aktualisieren. Zum Beispiel können eine oder mehrere Verlustfunktionen verwendet werden (z. B. kann eine Regressionsverlustfunktion wie L1- oder L2-Verlust für Regressionsaufgaben verwendet werden), um die Genauigkeit der Ausgabe(n) des/der Deep-Learning-Modells/Modelle mit der Ground Truth zu vergleichen, und die Parameter des/der Deep-Learning-Modells/Modelle können aktualisiert werden (z. B. unter Verwendung von Rückwärtsdurchläufen, Rückwärtspropagierung, Vorwärtsdurchläufen usw.), bis die Genauigkeit ein optimales oder akzeptables Niveau erreicht. In einigen Ausführungsformen, in denen das Deep-Learning-Modell mehrere Köpfe umfasst, können die mehreren Köpfe zusammen auf demselben Datensatz mit einem gemeinsamen Stamm trainiert werden. Auf diese Weise können sich die verschiedenen Köpfe (Aufgaben) gegenseitig beim Lernen helfen.

[0116] In einer beispielhaften Ausführungsform, in der das Deep-Learning-Modell einen Regressionskopf enthält, der eine Höhenkarte vorhersagt, kann das Deep-Learning-Modell lernen, Höhenkarten unter Verwendung von Ground-Truth-Höhenkarten und Ground-Truth-Segmentierungsmasken vorherzusagen. Beispielsweise kann eine Regressionsverlustfunktion wie L 1- oder L2-Verlust verwendet werden, um eine vorhergesagte Höhenkarte mit einer Ground-Truth-Höhenkarte zu vergleichen, und das Ergebnis kann mit einer Ground-Truth-Segmentierungsmaske multipliziert werden, die die zu verdichtende Fläche angibt, wodurch Aktualisierungen des/der Deep-Learning-Modelle(s) basierend auf Vorhersagen, die außerhalb des zu verdichtenden Bereichs auftreten, effektiv aufgehoben werden.

[0117] In einer anderen Ausführungsform, in der das Deep-Learning-Modell einen Regressionskopf, der eine Höhenkarte vorhersagt, und einen Konfidenzkopf umfasst, der eine der Höhenkarte entsprechende Konfidenzkarte vorhersagt, kann das Deep-Learning-Modell lernen, sowohl die Höhen- als auch die Konfidenzkarte aus den Ground-Truth-Höhenkarten der vorherzusagen. Beispielsweise kann eine Verlustfunktion verwendet werden, die die vorhergesagte Höhe mit der tatsächlichen Höhe vergleicht und basierend auf eines vorhergesagten Konfidenzwertes kompensiert. Ein Beispiel für eine solche Verlustfunktion kann gegeben sein durch:

$$L = \frac{\|y' - y\|^2}{2 * c^2} + \frac{1}{2} * \log c^2 \qquad \text{Gl. (6)}$$

wobei y eine vorhergesagte Höhe, y' eine entsprechende Ground-Truth-Höhe und c ein vorhergesagter Konfidenzwert ist, der der vorhergesagten Höhe entspricht. Wenn in diesem Beispiel die vorhergesagte Höhe wesentlich falsch ist (und ||y' - y|| daher groß ist), fördert die Minimierung dieser Verlustfunktion einen großen Wert von c. Daher kann in diesem

Beispiel ein großer Wert von c eine geringe Konfidenz anzeigen. Der logarithmische Term in dem durch Gleichung 6 gegebenen Beispielverlust verhindert, dass c unendlich groß wird. Eine Verlustfunktion wie diese kann verwendet werden, um ein Deep-Learning-Modell zu trainieren, um sowohl eine Höhenkarte als auch eine Konfidenzkarte vorherzusagen, ohne dass eine Ground-Truth-Konfidenzkarte erforderlich ist. Das/die Deep-Learning-Modell(e) können so trainiert werden, dass sie eine Verdichtung durchführen, indem sie eine Zuordnung zwischen spärlichen und dichten Darstellungen der 3D-Struktur lernen.

[0118] Nun bezugnehmend auf die **Fig. 14-16**, weist jeder Block der hier beschriebenen Verfahren 1400-1600 einen Rechenprozess auf, der mit einer beliebigen Kombination aus Hardware, Firmware und/oder Software durchgeführt werden kann. Beispielsweise können verschiedene Funktionen von einem Prozessor ausgeführt werden, der im Speicher gespeicherte Anweisungen ausführt. Die Verfahren können auch als computerverwendbare Anweisungen auf Computerspeichermedien gespeichert sein. Die Methoden können durch eine eigenständige Anwendung, einen Dienst oder einen gehosteten Dienst (eigenständig oder in Kombination mit einem anderen gehosteten Dienst) oder ein Plug-in für ein anderes Produkt bereitgestellt werden, um nur einige zu nennen. Auch wenn die Methoden 1400-1600 beispielhaft für ein Beispielsystem beschrieben werden, können diese Methoden zusätzlich oder alternativ von jedem beliebigen System oder jeder beliebigen Kombination von Systemen ausgeführt werden, einschließlich, aber nicht beschränkt auf die hier beschriebenen Systeme.

[0119] **Fig. 14** ist ein Flussdiagramm, das ein Verfahren 1400 zum Trainieren eines oder mehrerer neuronaler Netze (NNs) zeigt, um eine verdichtete Darstellung der 3D-Oberflächenstruktur unter Verwendung simulierter Bilddaten zu erzeugen, gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Das Verfahren 1400 umfasst in Block B 1402 den Zugriff auf simulierte Bilddaten und entsprechende Klassifizierungsdaten und Entfernungsdaten. Zum Beispiel kann die Simulatorkomponente 1010 der **Fig. 10** eine Simulation ausführen, um eine virtuelle Welt oder Umgebung (z. B. eine simulierte Umgebung) zu simulieren, und ein virtuelles Fahrzeug oder ein anderes Objekt kann innerhalb der simulierten Umgebung simuliert werden. Das virtuelle Fahrzeug oder Objekt kann eine beliebige Anzahl von Sensoren enthalten (z. B. virtuelle oder simulierte Sensoren), und für die Sensoren können virtuelle Sensordaten simuliert werden. So können Frames virtueller Sensordaten (z. B. virtuelle Bilddaten, die einem oder mehreren Sichtfeldern der virtuellen Kamera(s) eines virtuellen Fahrzeugs entsprechen) und entsprechende Segmentierungsmasken und Tiefenkarten basierend auf der simulierten Umgebung erzeugt werden.

[0120] Das Verfahren 1400 umfasst in Block B1404 die Erzeugung einer ersten Darstellung einer dreidimensionalen (3D) Oberflächenstruktur einer Straße, die durch die simulierten Bilddaten repräsentiert wird, mindestens teilweise basierend auf der simulierten Bilddaten. Beispielsweise kann der 3D-Strukturschätzer 1050 der **Fig. 10** eine spärliche Darstellung (z. B. eine spärliche Punktwolke, ein Projektionsbild) einer 3D-Oberflächenstruktur einer Straße erzeugen, die in einem gerenderten Bild (z. B. den Frames 1020) dargestellt wird, indem eine 3D-Strukturschätzung an dem gerenderten Bild durchgeführt wird.

[0121] Das Verfahren 1400 umfasst in Block B1406 das Erzeugen einer zweiten Darstellung der 3D-Oberflächenstruktur der Straße, die mindestens auf den Entfernungsdaten und den Klassifizierungsdaten basiert. Beispielsweise kann der 3D-Punktwolkengenerator 1060 der **Fig. 10** die Entfernungswerte aus der/den Tiefenkarte(n) 1040 umgekehrt in 3D-Weltkoordinaten projizieren, indem er die bekannte Position und Ausrichtung der virtuellen Kamera verwendet, relativ zu der die Entfernungswerte der Tiefenkarte(n) 1040 erzeugt wurden, der 3D-Punktwolkengenerator 1060 kann die Segmentierungsmaske (n) 1030 verwenden, um 3D-Punkte auf der interessierenden Oberfläche (z. B. einer Straßenoberfläche) herauszufiltern, und der Nachprozessor 1070 kann verwendet werden, um fehlende Werte aufzufüllen.

[0122] Das Verfahren 1400 umfasst in Block B1408 das Trainieren eines oder mehrerer neuronaler Netze (NNs), um eine verdichtete Darstellung der 3D-Oberflächenstruktur zu erzeugen, wobei die erste Darstellung der 3D-Oberflächenstruktur als Eingabetrainingsdaten und die zweite Darstellung der 3D-Oberflächenstruktur als Ground-Truth-Trainingsdaten verwendet werden.

[0123] **Fig. 15** ist ein Flussdiagramm, das ein Verfahren 1500 zur Erzeugung unvollständiger und Ground-Truth-Darstellungen einer synthetischen 3D-Straßenoberfläche für einen Trainingsdatensatz gemäß einigen Ausführungsformen der vorliegenden Offenbarung zeigt. Das Verfahren 1500 umfasst im Block B1502 die Erzeugung einer Darstellung einer longitudinalen dreidimensionalen (3D) Kurve, die eine synthetische Straße darstellt. In Bezug auf **Fig. 11** kann beispielsweise eine Darstellung der longitudinalen Kurve l mit longitudinalen Werten erzeugt werden, die einen gewünschten Wahrnehmungsbereich für einen Deep-Learning-Modell-Oberflächenschätzer darstellen, mit lateralen Werten, die als Polynom zweiter Ordnung der longitudinalen Werte berechnet werden, und mit Höhenwerten, die als

lineare Kombination von Fourier-Basen berechnet werden.

[0124] Das Verfahren 1500 umfasst im Block B1504 für jeden Punkt eines oder mehrerer Punkte auf der longitudinalen 3D-Kurve die Erweiterung des Punktes zu einer lateralen 3D-Kurve durch den Punkt. In Bezug auf **Fig. 11** kann beispielsweise jeder gegebene Punkt (z. B. jeder Punkt) auf der longitudinalen 3D-Kurve l in eine entsprechende laterale 3D-Kurve erweitert werden, die durch die Kurven q_j dargestellt wird. In einigen Ausführungsformen kann ein Parameter α definiert werden, um den Winkel zwischen einer synthetischen 3D-Oberfläche (z. B. der synthetischen 3D-Straßenoberfläche) und der Oberfläche z = 0 (z. B. der Bodenebene) zu bezeichnen, und verschiedene Werte von α können abgetastet werden, um verschiedene laterale Oberflächenneigungen an verschiedenen Punkten der longitudinalen 3D-Kurve und/oder für verschiedene synthetische 3D-Kurven zu simulieren. So kann jeder beliebige Punkt (z. B. jeder Punkt) auf der longitudinalen 3D-Kurve l zu einer lateralen 3D-Kurve q_j erweitert werden, die durch p_j senkrecht zur Kurve l bei p_j verläuft und einen Winkel α relativ zur Oberfläche z = 0 aufweist.

[0125] Das Verfahren 1500 umfasst in Block B1506 die Erzeugung einer Ground-Truth-Darstellung einer synthetischen 3D-Straßenoberfläche der synthetischen Straße, die mindestens auf der lateralen 3D-Kurve für zwei oder mehr Punkte auf der longitudinalen 3D-Kurve basiert. In Bezug auf **Fig. 11** kann zum Beispiel jede gegebene laterale 3D-Kurve q_j m-mal abgetastet werden, um einen entsprechenden 3D-Punkt auf der longitudinalen 3D-Kurve l zu einem Satz von 3D-Punkten zu erweitern, um verschiedene Straßenbreiten an verschiedenen Punkten auf der longitudinalen 3D-Kurve zu simulieren. Der Vorgang kann für jeden beliebigen Punkt (z. B. jeden Punkt) der longitudinalen 3D-Kurve l wiederholt werden, um eine dichte 3D-Punktwolke zu erzeugen, die projiziert werden kann, um ein Ground-Truth-Projektionsbild (z. B. eine 2D-Ground-Truth-Höhenkarte) zu bilden.

[0126] Das Verfahren 1500 umfasst in Block B1508 die Erzeugung einer unvollständigen Darstellung der synthetischen 3D-Straßenoberfläche, die mindestens auf der Ground-Truth-Darstellung der synthetischen 3D-Straßenoberfläche basiert. Beispielsweise kann ein Muster, das durch Vbinäre Karten der Größe HxWdargestellt wird, erzeugt werden, indem eine 3D-Schätzung an realen Daten durchgeführt wird und eine Darstellung codiert wird, welche Pixel bei der Durchführung der 3D-Schätzung aus aufgenommenen Bildern beobachtet und nicht beobachtet werden. So kann eine der N binären Karten zufällig abgetastet und auf die Ground-Truth-Darstellung der synthetischen 3D-Straßenoberfläche angewendet werden (z. B. durch elementweise Multiplikation), um eine entsprechende unvollständige Darstellung der synthetischen 3D-Straßenoberfläche zu erzeugen.

[0127] In Block B1510 werden die unvollständige Darstellung und die Ground-Truth-Darstellung in einen Trainingsdatensatz aufgenommen.

[0128] **Fig. 16** ist ein Flussdiagramm, das ein Verfahren 1600 zum Trainieren eines oder mehrerer neuronaler Netze (NNs) zeigt, um eine verdichtete Darstellung der 3D-Oberflächenstruktur unter Verwendung von Bilddaten und LiDAR-Daten zu erzeugen, die während einer Erfassungssitzung erfasst wurden, gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Das Verfahren 1600 umfasst in Block B 1602 den Zugriff auf Bilddaten und LiDAR-Daten, die während einer Erfassungssitzung in einer Umgebung erfasst wurden. Beispielsweise kann ein Datensammelfahrzeug mit einer oder mehreren Kamera(s) und LiDAR-Sensor(en) ausgestattet sein, und die Aufzeichnungs-Engine 1205 der **Fig. 12** kann Sensordaten aufzeichnen, während das Fahrzeug durch reale (z. B. physikalische) Umgebungen fährt.

[0129] Das Verfahren 1600 umfasst in Block B 1604 das Erzeugen einer unvollständigen Darstellung einer dreidimensionalen, 3D, Oberflächenstruktur der Straße in der Umgebung, mindestens basierend auf der Bilddaten. Beispielsweise kann der 3D-Strukturschätzer 1220 der **Fig. 12** für beliebige gegebene Frames 1210 von Bilddaten eine 3D-Oberflächenstruktur einer Straße aus den Frames 1210 schätzen (z. B. wie oben in Bezug auf den 3D-Strukturschätzer 105 der **Fig. 1** und/oder den 3D-Strukturschätzer 1050 der **Fig. 10** beschrieben). In einigen Ausführungsformen kann der 3D-Strukturschätzer 1220 die Segmentierungsmaske(n) 1230 verwenden, um Punkte aus einer geschätzten 3D-Struktur auszuwählen, die zu einer durch die Segmentierungsmaske repräsentierten Klasse gehören (z. B. Punkte, die zu einer 3D-Straßenoberfläche gehören). In einigen Ausführungsformen können die resultierenden Punkte projiziert werden, um ein Projektionsbild zu bilden (z. B. eine 2D-Höhenkarte).

[0130] Das Verfahren 1600 umfasst in Block 1606 das Erzeugen einer zweiten Darstellung der 3D-Oberflächenstruktur der Straße, die mindestens auf einer Kennzeichnung der LiDAR-Daten basiert. In Bezug auf **Fig. 12** kann der Vorprozessor 1240 beispielsweise eine oder mehrere Verarbeitungsoperationen an den LiDAR-Daten 1215 vor der Kennzeichnung durchführen, wie z. B. zeitliche Glättung, Ausreißerentfernung, Triangulation und/oder Akkumulation von mehreren LiDAR-Sensoren. In einigen Ausführungsformen kann der Aligner 1250 die LiDAR-Daten zeitlich und/oder räumlich mit den entsprechenden Frames 1210 der Bilddaten ausrichten.

In einem Ausführungsbeispiel kann die Annotationskomponente 1260 die ausgerichteten LiDAR-Daten und Bilddaten in einer Annotationsszene einem menschlichen Kennzeichner präsentieren, Eingaben akzeptieren, die Ground-Truth-Annotationen angeben, die Punkte auf einer interessierenden Oberfläche identifizieren, und eine Darstellung der identifizierten Punkte erzeugen, die der Ansicht, Größe und Dimensionalität der Ausgabe(n) des/der zu trainierenden Deep-Learning-Modells/Modelle entspricht.

[0131] Das Verfahren 1600 umfasst in Block B1608 das Trainieren eines oder mehrerer neuronaler Netze (NNs), um eine verdichtete Darstellung der 3D-Oberflächenstruktur zu erzeugen, wobei die unvollständige Darstellung der 3D-Oberflächenstruktur als Eingabetrainingsdaten und die zweite Darstellung der 3D-Oberflächenstruktur als Ground-Truth-Trainingsdaten verwendet werden.

Beispielhaftes autonomes Fahrzeug

[0132] **Fig. 17A** ist eine Darstellung eines Beispiels für ein autonomes Fahrzeug 1700 gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Das autonome Fahrzeug 1700 (hier alternativ als „Fahrzeug 1700“ bezeichnet) kann ohne Einschränkung ein Personenfahrzeug sein, wie z. B. ein Pkw, ein Lkw, ein Bus, ein First-Responder-Fahrzeug, ein Shuttle, ein elektrisches oder motorisiertes Fahrrad, ein Motorrad, ein Feuerwehrauto, ein Polizeifahrzeug, ein Krankenwagen, ein Boot, ein Baufahrzeug, ein Unterwasserfahrzeug, eine Drohne, ein an einen Anhänger gekoppeltes Fahrzeug und/oder eine andere Art von Fahrzeug (z. B. das unbemannt ist und/oder einen oder mehrere Fahrgäste aufnimmt). Autonome Fahrzeuge werden im Allgemeinen in Form von Automatisierungsstufen beschrieben, die von der National Highway Traffic Safety Administration (NHTSA), einer Abteilung des US-Verkehrsministeriums, und der Society of Automotive Engineers (SAE) „Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles“ (Standard Nr. J3016-201806, veröffentlicht am 15. Juni 2018, Standard Nr. J3016-201609, veröffentlicht am 30. September 2016, sowie frühere und zukünftige Versionen dieses Standards) definiert werden. Das Fahrzeug 1700 kann in der Lage sein, Funktionen gemäß einer oder mehrerer der Stufen 3 bis 5 der autonomen Fahrstufen auszuführen. Zum Beispiel kann das Fahrzeug 1700 je nach Ausführungsform bedingt automatisiert (Stufe 3), hochautomatisiert (Stufe 4) und/oder vollständig automatisiert (Stufe 5) sein.

[0133] Das Fahrzeug 1700 kann Komponenten wie ein Fahrgestell, eine Fahrzeugkarosserie, Räder (z. B. 2, 4, 6, 8, 18 usw.), Reifen, Achsen und andere Komponenten eines Fahrzeugs umfassen. Das Fahrzeug 1700 kann ein Antriebssystem 1750 enthalten, wie z. B. einen Verbrennungsmotor, ein Hybrid-Elektrokraftwerk, einen reinen Elektromotor und/oder einen anderen Antriebssystemtyp. Das Antriebssystem 1750 kann mit einem Antriebsstrang des Fahrzeugs 1700 verbunden sein, der ein Getriebe umfassen kann, um den Antrieb des Fahrzeugs 1700 zu ermöglichen. Das Antriebssystem 1750 kann in Reaktion auf den Empfang von Signalen von der Drosselklappe/Beschleunigungsvorrichtung 1752 gesteuert werden.

[0134] Ein Lenksystem 1754, das ein Lenkrad umfassen kann, kann verwendet werden, um das Fahrzeug 1700 zu lenken (z. B. entlang eines gewünschten Weges oder einer Route), wenn das Antriebssystem 1750 in Betrieb ist (z. B. wenn das Fahrzeug in Bewegung ist). Das Lenksystem 1754 kann Signale von einem Lenkaktor 1756 empfangen. Das Lenkrad kann optional für die vollständige Automatisierung (Stufe 5) eingesetzt werden.

[0135] Das Bremssensorsystem 1746 kann verwendet werden, um die Fahrzeugbremsen als Reaktion auf den Empfang von Signalen von den Bremsaktuatoren 1748 und/oder Bremssensoren zu betätigen.

[0136] Ein oder mehrere Controller 1736, die ein oder mehrere System-on-Chips (SoCs) 1704 (**Fig. 17C**) und/oder GPU(s) enthalten können, können Signale (z. B. repräsentativ für Befehle) an eine oder mehrere Komponenten und/oder Systeme des Fahrzeugs 1700 senden. Beispielsweise können die Controller Signale zur Betätigung der Fahrzeugbremsen über einen oder mehrere Bremsaktuatoren 1748, zur Betätigung des Lenksystems 1754 über einen oder mehrere Lenkaktuatoren 1756 und zur Betätigung des Antriebssystems 1750 über einen oder mehrere Drosselklappen-/Gaspedale 1752 senden. Die Controller 1736 können eine oder mehrere eingebaute (z. B. integrierte) Rechenvorrichtungen (z. B. Supercomputer) umfassen, die Sensorsignale verarbeiten und Betriebsbefehle ausgeben (z. B. Signale, die Befehle darstellen), um autonomes Fahren zu ermöglichen und/oder einen menschlichen Fahrer beim Führen des Fahrzeugs 1700 zu unterstützen. Die Controller 1736 können einen ersten Controller 1736 für autonome Fahrfunktionen, einen zweiten Controller 1736 für funktionale Sicherheitsfunktionen, einen dritten Controller 1736 für Funktionen der künstlichen Intelligenz (z. B. Computervision), einen vierten Controller 1736 für Infotainment-Funktionen, einen fünften Controller 1736 für Redundanz unter Notfallbedingungen und/oder andere Controller umfassen. In einigen Beispielen kann ein einziger Controller 1736 zwei oder mehr der oben genannten Funktionen übernehmen, zwei oder mehr Controller 1736 können eine einzige Funktion übernehmen und/oder eine beliebige Kombination davon.

[0137] Der/die Controller 1736 können die Signale zur Steuerung einer oder mehrerer Komponenten und/oder Systeme des Fahrzeugs 1700 als Reaktion auf Sensordaten bereitstellen, die von einem oder mehreren Sensoren (z. B. Sensoreingaben) empfangen werden. Die Sensordaten können beispielsweise und ohne Einschränkung von (einem) Sensor (en) des globalen Navigationssatellitensystems 1758 (z. B. Global Positioning System-Sensor(en)), RADAR-Sensor(en) 1760, Ultraschallsensor(en) 1762, LIDAR-Sensor(en) 1764, Trägheitsmesseinheits-(IMU)-Sensor(en) 1766 (z. B., Beschleunigungsmesser, Gyroskop(e), Magnetkompass(e), Magnetometer usw.), Mikrofon(e) 1796, Stereokamera(s) 1768, Weitwinkelkamera(s) 1770 (z. B., Fischaugenkameras), Infrarotkamera(n) 1772, Surround-Kamera(n) 1774 (z. B. 360-Grad-Kameras), Fern- und/oder Mittelbereichskamera(n) 1798, Geschwindigkeitssensor(en) 1744 (z. B. zur Messung der Geschwindigkeit des Fahrzeugs 1700), Vibrationssensor(en) 1742, Lenksensor(en) 1740, Bremssensor(en) (z. B. als Teil des Bremssensorsystems 1746) und/oder andere Sensortypen.

[0138] Ein oder mehrere Controller 1736 können Eingaben (z. B. in Form von Eingabedaten) von einem Kombiinstrument 1732 des Fahrzeugs 1700 empfangen und Ausgaben (z. B. in Form von Ausgabedaten, Anzeigedaten usw.) über eine Mensch-Maschine-Schnittstelle (HMI)-Anzeige 1734, einen akustischen Melder, einen Lautsprecher und/oder über andere Komponenten des Fahrzeugs 1700 liefern. Die Ausgaben können Informationen wie Fahrzeuggeschwindigkeit, Geschwindigkeit, Zeit, Kartendaten (z. B. die HD-Karte 1722 der **Fig. 17C**), Standortdaten (z. B. der Standort des Fahrzeugs 1700, wie auf einer Karte), Richtung, Standort anderer Fahrzeuge (z. B. ein Belegungsraster), Informationen über Objekte und den Status von Objekten, wie von dem/den Controller(n) 1736 wahrgenommen, usw. umfassen. Beispielsweise kann die HMI-Anzeige 1734 Informationen über das Vorhandensein eines oder mehrerer Objekte (z. B. ein Straßenschild, ein Warnschild, eine sich ändernde Ampel usw.) und/oder Informationen über Fahrmanöver anzeigen, die das Fahrzeug durchgeführt hat, gerade durchführt oder durchführen wird (z. B. Spurwechsel jetzt, Ausfahrt 34B in zwei Meilen usw.).

[0139] Das Fahrzeug 1700 umfasst ferner eine Netzwerkschnittstelle 1724, die eine oder mehrere drahtlose Antenne(n) 1726 und/oder Modem(e) zur Kommunikation über ein oder mehrere Netzwerke verwenden kann. Zum Beispiel kann die Netzwerkschnittstelle 1724 in der Lage sein, über LTE, WCDMA, UMTS, GSM, CDMA2000, etc. zu kommunizieren. Die drahtlose(n) Antenne(n) 1726 können auch die Kommunikation zwischen Objekten in der Umgebung (z. B. Fahrzeuge, mobile Geräte usw.) über lokale Netzwerke wie Bluetooth, Bluetooth LE, Z-Wave, ZigBee usw. und/oder Low Power Wide Area Networks (LPWANs) wie LoRaWAN, SigFox usw. ermöglichen.

[0140] **Fig. 17B** ist ein Beispiel für Kamerapositionen und Sichtfelder für das autonome Fahrzeug 1700 der **Fig. 17A**, gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Die Kameras und die jeweiligen Sichtfelder stellen ein Ausführungsbeispiel dar und sind nicht als einschränkend zu betrachten. Zum Beispiel können zusätzliche und/oder alternative Kameras enthalten sein und/oder die Kameras können an verschiedenen Stellen des Fahrzeugs 1700 angeordnet sein.

[0141] Die Kameratypen für die Kameras können unter anderem Digitalkameras sein, die für die Verwendung mit den Komponenten und/oder Systemen des Fahrzeugs 1700 angepasst werden können. Die Kamera(s) können mit der Sicherheitsstufe B (ASIL) und/oder einer anderen ASIL betrieben werden. Die Kameratypen können je nach Ausführungsform eine beliebige Bildaufnahmerate haben, z. B. 60 Bilder pro Sekunde (fps), 120 fps, 240 fps usw. Die Kameras können Rolling Shutter, Global Shutter, einen anderen Verschlusstyp oder eine Kombination davon verwenden. In einigen Beispielen kann die Farbfilteranordnung eine Rot-Klar-Klar-Klar-Farbfilteranordnung (RCCC), eine Rot-Klar-Klar-Blau-Farbfilteranordnung (RCCB), eine Rot-Blau-Grün-Klar-Farbfilteranordnung (RBGC), eine Foveon X3-Farbfilteranordnung, eine Bayer-Sensor-Farbfilteranordnung (RGGB), eine Monochromsensor-Farbfilteranordnung und/oder eine andere Art von Farbfilteranordnung umfassen. In einigen Ausführungsformen können Kameras mit klaren Pixeln, wie z. B. Kameras mit einer RCCC-, einer RCCB- und/oder einer RBGC-Farbfilteranordnung, verwendet werden, um die Lichtempfindlichkeit zu erhöhen.

[0142] In einigen Beispielen können eine oder mehrere der Kameras verwendet werden, um erweiterte Fahrerassistenzsysteme (ADAS) auszuführen (z. B. als Teil einer redundanten oder ausfallsicheren Konstruktion). So kann beispielsweise eine Multifunktions-Monokamera installiert werden, um Funktionen wie Spurhalteassistent, Verkehrszeichenassistent und intelligente Scheinwerfersteuerung bereitzustellen. Eine oder mehrere der Kameras (z. B. alle Kameras) können gleichzeitig Bilddaten (z. B. Video) aufzeichnen und liefern.

[0143] Eine oder mehrere Kameras können in einer Montagevorrichtung, z. B. einer kundenspezifischen (3-D-gedruckten) Vorrichtung, montiert werden, um Streulicht und Reflexionen aus dem Fahrzeuginneren (z. B. Reflexionen vom Armaturenbrett, die sich in den Windschutzscheibenspiegeln spiegeln) auszuschalten, die die Bilddatenerfassung der Kamera

beeinträchtigen könnten. Bei der Montage von Außenspiegeln können die Außenspiegel kundenspezifisch dreidimensional gedruckt werden, so dass die Kameramontageplatte der Form des Außenspiegels entspricht. In einigen Beispielen können die Kameras in den Außenspiegel integriert werden. Bei Seitenkameras können die Kameras auch in die vier Säulen an jeder Ecke der Kabine integriert werden.

[0144] Kameras mit einem Sichtfeld, das Teile der Umgebung vor dem Fahrzeug 1700 einschließt (z. B. nach vorne gerichtete Kameras), können für die Umgebungsansicht verwendet werden, um bei der Erkennung von nach vorne gerichteten Wegen und Hindernissen zu helfen, sowie mit Hilfe von einem oder mehreren Controllern 1736 und/oder Steuer-SoCs Informationen bereitzustellen, die für die Erstellung eines Belegungsrasters und/oder die Bestimmung der bevorzugten Fahrzeugwege entscheidend sind. Nach vorne gerichtete Kameras können verwendet werden, um viele der gleichen ADAS-Funktionen wie LIDAR auszuführen, einschließlich Notbremsung, Fußgängererkennung und Kollisionsvermeidung. Nach vorne gerichtete Kameras können auch für ADAS-Funktionen und -Systeme wie Spurverlassenswarnung (LDW), autonome Geschwindigkeitsregelung (ACC) und/oder andere Funktionen wie Verkehrszeichenerkennung verwendet werden.

[0145] In einer nach vorne gerichteten Konfiguration kann eine Vielzahl von Kameras verwendet werden, z. B. eine monokulare Kameraplattform, die einen CMOS-Farbbildsensor (Komplementär-Metalloxid-Halbleiter) enthält. Ein weiteres Beispiel sind eine oder mehrere Weitwinkelkameras 1770, die verwendet werden können, um Objekte zu erkennen, die von der Peripherie her ins Blickfeld kommen (z. B. Fußgänger, kreuzenden Verkehr oder Fahrräder). Obwohl in **Fig. 17B** nur eine Weitwinkelkamera abgebildet ist, kann sich eine beliebige Anzahl von Weitwinkelkameras 1770 am Fahrzeug 1700 befinden. Darüber hinaus können die Weitwinkelkamera (s) 1798 (z. B. ein Weitwinkel-Stereokamerapaar) zur tiefenbasierten Objekterkennung verwendet werden, insbesondere für Objekte, für die ein neuronales Netz noch nicht trainiert wurde. Die Langstreckenkamera(s) 1798 können auch zur Objekterkennung und - klassifizierung sowie zur grundlegenden Objektverfolgung eingesetzt werden.

[0146] Eine oder mehrere Stereokameras 1768 können auch in einer nach vorne gerichteten Konfiguration enthalten sein. Die Stereokamera(s) 1768 können eine integrierte Steuereinheit mit einer skalierbaren Verarbeitungseinheit umfassen, die eine programmierbare Logik (FPGA) und einen Multicore-Mikroprozessor mit integrierter CAN- oder Ethernet-Schnittstelle auf einem einzigen Chip bereitstellen kann. Mit einer solchen Einheit kann eine 3D-Karte der Fahrzeugumgebung erstellt werden, einschließlich einer Entfernungsschätzung für alle Punkte im Bild. Eine alternative Stereokamera (s) 1768 kann einen kompakten Stereosicht-Sensor bzw. kompakte Stereosicht-Sensoren umfassen, der bzw. die zwei Kameralinsen (je eine links und rechts) und einen Bildverarbeitungschip enthält bzw. enthalten, der die Entfernung zwischen dem Fahrzeug und dem Zielobjekt messen und die erzeugten Informationen (z. B. Metadaten) zur Aktivierung der autonomen Notbrems- und Spurhaltewarnfunktionen verwenden kann. Zusätzlich oder alternativ zu den hier beschriebenen Stereokameras können auch andere Typen von Stereokamera(s) 1768 verwendet werden.

[0147] Kameras mit einem Sichtfeld, das Teile der Umgebung seitlich des Fahrzeugs 1700 einschließt (z. B. Seitenkameras), können für die Umgebungsansicht verwendet werden und Informationen liefern, die zur Erstellung und Aktualisierung des Belegungsrasters sowie zur Erzeugung von Kollisionswarnungen bei Seitenaufprall verwendet werden. Beispielsweise können die Surround-Kamera(s) 1774 (z. B. vier Surround-Kameras 1774, wie in **Fig. 17B** dargestellt) am Fahrzeug 1700 positioniert werden. Die Surround-Kamera(s) 1774 können Breitbildkamera (s) 1770, Fischaugenkamera(s), 360-Grad-Kamera (s) und/oder Ähnliches umfassen. Vier Beispiele: Vier Fischaugenkameras können an der Vorderseite, am Heck und an den Seiten des Fahrzeugs angebracht werden. In einer alternativen Anordnung kann das Fahrzeug drei Surround-Kameras 1774 (z. B. links, rechts und hinten) verwenden und eine oder mehrere andere Kamera(s) (z. B. eine nach vorne gerichtete Kamera) als vierte Surround-Kamera nutzen.

[0148] Kameras mit einem Sichtfeld, das Teile der Umgebung hinter dem Fahrzeug 1700 einschließt (z. B. Rückfahrkameras), können für die Einparkhilfe, die Umgebungsansicht, Heckkollisionswarnungen und das Erstellen und Aktualisieren des Belegungsrasters verwendet werden. Es kann eine Vielzahl von Kameras verwendet werden, einschließlich, aber nicht beschränkt auf Kameras, die auch als nach vorne gerichtete Kamera(s) geeignet sind (z. B. Fern- und/oder Mittelbereichskamera(s) 1798, Stereokamera(s) 1768, Infrarotkamera(s) 1772 usw.), wie hier beschrieben.

[0149] Fig. 17C ist ein Blockdiagramm einer beispielhaften Systemarchitektur für das beispielhafte autonome Fahrzeug 1700 der **Fig. 17A**, gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Es sollte verstanden werden, dass diese und andere hier beschriebene Anordnungen nur als Beispiele dargestellt werden. Andere Anordnungen und Elemente (z. B. Maschinen, Schnittstellen, Funktionen, Anordnungen, Gruppierungen von Funktionen

usw.) können zusätzlich zu oder anstelle der dargestellten verwendet werden, und einige Elemente können ganz weggelassen werden. Außerdem sind viele der hier beschriebenen Elemente funktionale Einheiten, die als einzelne oder verteilte Komponenten oder in Verbindung mit anderen Komponenten und in jeder geeigneten Kombination und an jedem geeigneten Ort implementiert werden können. Verschiedene hier beschriebene Funktionen, die von Einheiten ausgeführt werden, können von Hardware, Firmware und/oder Software ausgeführt werden. Beispielsweise können verschiedene Funktionen von einem Prozessor ausgeführt werden, der im Speicher gespeicherte Anweisungen ausführt.

[0150] Alle Komponenten, Merkmale und Systeme des Fahrzeugs 1700 in **Fig. 17C** sind als über den Bus 1702 verbunden dargestellt. Der Bus 1702 kann eine Controller Area Network (CAN)-Datenschnittstelle enthalten (hier alternativ als „CAN-Bus“ bezeichnet). Bei einem CAN-Bus kann es sich um ein Netzwerk innerhalb des Fahrzeugs 1700 handeln, das zur Steuerung verschiedener Merkmale und Funktionen des Fahrzeugs 1700 verwendet wird, z. B. zur Betätigung von Bremsen, Beschleunigung, Bremsen, Lenkung, Scheibenwischern usw. Ein CAN-Bus kann so konfiguriert sein, dass er Dutzende oder sogar Hunderte von Knoten hat, von denen jeder seine eigene eindeutige Kennung hat (z. B. eine CAN-ID). Der CAN-Bus kann ausgelesen werden, um den Lenkradwinkel, die Fahrgeschwindigkeit, die Motordrehzahl (RPM), die Tastenpositionen und/oder andere Fahrzeugstatusanzeigen zu ermitteln. Der CAN-Bus kann ASIL B-konform sein.

[0151] Obwohl der Bus 1702 hier als CAN-Bus beschrieben wird, ist dies nicht als Einschränkung zu verstehen. Beispielsweise können zusätzlich oder alternativ zum CAN-Bus auch FlexRay und/oder Ethernet verwendet werden. Obwohl eine einzelne Leitung zur Darstellung des Busses 1702 verwendet wird, ist dies nicht als Einschränkung zu verstehen. So kann es beispielsweise eine beliebige Anzahl von Bussen 1702 geben, die einen oder mehrere CAN-Busse, einen oder mehrere FlexRay-Busse, einen oder mehrere Ethernet-Busse und/oder einen oder mehrere andere Arten von Bussen mit einem anderen Protokoll umfassen können. In einigen Beispielen können zwei oder mehr Busse 1702 verwendet werden, um unterschiedliche Funktionen auszuführen, und/oder sie können zur Redundanz verwendet werden. So kann beispielsweise ein erster Bus 1702 für die Kollisionsvermeidungsfunktionalität und ein zweiter Bus 1702 für die Betätigungssteuerung verwendet werden. In jedem Beispiel kann jeder Bus 1702 mit einer beliebigen Komponente des Fahrzeugs 1700 kommunizieren, und zwei oder mehr Busse 1702 können mit denselben Komponenten kommunizieren. In einigen Beispielen kann jeder SoC 1704, jeder Controller 1736 und/oder jeder Computer innerhalb des Fahrzeugs Zugriff auf dieselben Eingangsdaten haben (z. B. Eingaben von Sensoren des Fahrzeugs 1700) und mit einem gemeinsamen Bus, wie dem CAN-Bus, verbunden sein.

[0152] Das Fahrzeug 1700 kann ein oder mehrere Controller 1736 enthalten, wie sie hier in Bezug auf **Fig. 17A** beschrieben sind. Die Controller 1736 können für eine Vielzahl von Funktionen verwendet werden. Die Controller 1736 können mit den verschiedenen anderen Komponenten und Systemen des Fahrzeugs 1700 gekoppelt werden und können zur Steuerung des Fahrzeugs 1700, zur künstlichen Intelligenz des Fahrzeugs 1700, zum Infotainment für das Fahrzeug 1700 und/oder ähnlichem verwendet werden.

[0153] Das Fahrzeug 1700 kann ein oder mehrere System(e) auf einem Chip (SoC) 1704 enthalten. Der SoC 1704 kann CPU(s) 1706, GPU(s) 1708, Prozessor(en) 1710, Cache(s) 1712, Beschleuniger 1714, Datenspeicher 1716 und/oder andere nicht dargestellte Komponenten und Merkmale umfassen. Der/die SoC(s) 1704 können zur Steuerung des Fahrzeugs 1700 in einer Vielzahl von Plattformen und Systemen verwendet werden. Beispielsweise können die SoC(s) 1704 in einem System (z. B. dem System des Fahrzeugs 1700) mit einer HD-Karte 1722 kombiniert werden, die Kartenauffrischungen und/oder -aktualisierungen über eine Netzwerkschnittstelle 1724 von einem oder mehreren Servern (z. B. Server(n) 1778 der **Fig. 17D**) erhalten kann.

[0154] Die CPU(s) 1706 können einen CPU-Cluster oder CPU-Komplex (hier auch als „CCPLEX“ bezeichnet) umfassen. Die CPU(s) 1706 können mehrere Kerne und/oder L2-Caches enthalten. In einigen Ausführungsformen können die CPU(s) 1706 beispielsweise acht Kerne in einer kohärenten Multiprozessorkonfiguration umfassen. In einigen Ausführungsformen können die CPU(s) 1706 vier Dual-Core-Cluster umfassen, wobei jeder Cluster über einen dedizierten L2-Cache verfügt (z. B. einen 2 MB L2-Cache). Die CPU(s) 1706 (z. B. der CCPLEX) können so konfiguriert werden, dass sie den gleichzeitigen Clusterbetrieb unterstützen, so dass eine beliebige Kombination der Cluster der CPU(s) 1706 zu einem bestimmten Zeitpunkt aktiv sein kann.

[0155] Die CPU(s) 1706 können Energieverwaltungsfunktionen implementieren, die eines oder mehrere der folgenden Merkmale umfassen: einzelne Hardwareblöcke können im Leerlauf automatisch taktgesteuert werden, um dynamischen Strom zu sparen; jeder Kerntakt kann gesteuert werden, wenn der Kern aufgrund der Ausführung von WFI/WFE-Befehlen nicht aktiv Befehle ausführt;

jeder Kern kann unabhängig stromgesteuert sein; jeder Kern-Cluster kann unabhängig taktgesteuert sein, wenn alle Kerne taktgesteuert oder stromgesteuert sind; und/oder jeder Kern-Cluster kann unabhängig stromgesteuert sein, wenn alle Kerne stromgesteuert sind. Die CPU(s) 1706 können darüber hinaus einen erweiterten Algorithmus zur Verwaltung von Energiezuständen implementieren, bei dem zulässige Energiezustände und erwartete Aufwachzeiten festgelegt werden und die Hardware/der Mikrocode den besten Energiezustand für den Kern, den Cluster und CCPLEX bestimmt. Die Prozessorkerne können vereinfachte Sequenzen für den Eintritt in den Energiezustand in Software unterstützen, wobei die Arbeit an den Mikrocode ausgelagert wird.

[0156] Die GPUs 1708 können eine integrierte GPU sein (hier alternativ als „iGPU“ bezeichnet). Die GPU(s) 1708 können programmierbar und für parallele Arbeitslasten effizient sein. Die GPU(s) 1708 können in einigen Beispielen einen erweiterten Tensor-Befehlssatz verwenden. Die GPU(s) 1708 können einen oder mehrere Streaming-Mikroprozessoren enthalten, wobei jeder Streaming-Mikroprozessor einen L1-Cache (z. B. einen L1-Cache mit mindestens 96 KB Speicherkapazität) enthalten kann und zwei oder mehr der Streaming-Mikroprozessoren sich einen L2-Cache (z. B. einen L2-Cache mit 512 KB Speicherkapazität) teilen können. In einigen Ausführungsformen können die GPU(s) 1708 mindestens acht Streaming-Mikroprozessoren umfassen. Die GPU(s) 1708 können eine oder mehrere Programmierschnittstellen (API(s)) für Berechnungen verwenden. Darüber hinaus können die GPU(s) 1708 eine oder mehrere parallele Rechenplattformen und/oder Programmiermodelle (z. B. CUDA von NVIDIA) verwenden.

[0157] Die GPUs 1708 können für die beste Leistung in Automobil- und eingebetteten Anwendungsfällen optimiert werden. Die GPU(s) 1708 können beispielsweise auf einem Fin-Feldeffekttransistor (FinFET) hergestellt werden. Dies ist jedoch nicht als Einschränkung zu verstehen, und die GPU(s) 1708 können auch mit anderen Halbleiterfertigungsverfahren hergestellt werden. Jeder Streaming-Mikroprozessor kann eine Reihe von gemischtpräzisen Rechenkernen enthalten, die in mehrere Blöcke unterteilt sind. So können beispielsweise 64 PF32-Kerne und 32 PF64-Keme in vier Verarbeitungsblöcke unterteilt werden, ohne darauf beschränkt zu sein. In einem solchen Beispiel können jedem Verarbeitungsblock 16 FP32-Keme, 8 FP64-Kerne, 16 INT32-Keme, zwei NVIDIA TENSOR COREs mit gemischter Präzision für Deep-Learning-Matrixarithmetik, ein L0-Befehlscache, ein Warp-Scheduler, eine Dispatch-Einheit und/oder eine 64-KB-Registerdatei zugewiesen werden. Darüber hinaus können die Streaming-Mikroprozessoren unabhängige parallele Ganzzahl- und Gleitkommadatenwege enthalten, um eine effiziente Ausführung von Arbeitslasten mit einer Mischung aus Berechnungen und Adressierungsberechnungen zu ermöglichen. Die Streaming-Mikroprozessoren können eine unabhängige Thread-Planungsfunktion enthalten, um eine feinere Synchronisierung und Zusammenarbeit zwischen parallelen Threads zu ermöglichen. Die Streaming-Mikroprozessoren können einen kombinierten L1-Datencache und eine gemeinsame Speichereinheit enthalten, um die Leistung zu verbessern und gleichzeitig die Programmierung zu vereinfachen.

[0158] Die GPUs 1708 können einen Speicher mit hoher Bandbreite (HBM) und/oder ein 16-GB-HBM2-Speicher-Subsystem umfassen, um in einigen Beispielen eine Spitzen-Speicherbandbreite von etwa 900 GB/Sekunde bereitzustellen. In einigen Beispielen kann zusätzlich oder alternativ zum HBM-Speicher ein synchroner Grafik-Direktzugriffsspeicher (SGRAM) verwendet werden, beispielsweise ein synchroner Grafik-Doppeldatenraten-Direktzugriffsspeicher vom Typ 5 (GDDR5).

[0159] Die GPU(s) 1708 können eine Unified-Memory-Technologie mit Zugriffszählern enthalten, um eine genauere Migration von Speicherseiten zu dem Prozessor zu ermöglichen, der am häufigsten auf sie zugreift, wodurch die Effizienz von Speicherbereichen verbessert wird, die von den Prozessoren gemeinsam genutzt werden. In einigen Beispielen kann die Unterstützung von Adressübersetzungsdiensten (ATS) verwendet werden, damit die GPU(s) 1708 direkt auf die Seitentabellen der CPU(s) 1706 zugreifen kann. In solchen Beispielen kann eine Adressübersetzungsanforderung an die CPU(s) 1706 übertragen werden, wenn die Speicherverwaltungseinheit (MMU) der GPU(s) 1708 einen Fehler feststellt. Als Antwort darauf kann die CPU(s) 1706 in ihren Seitentabellen nach der virtuell-physikalischen Zuordnung für die Adresse suchen und die Übersetzung zurück an die GPU(s) 1708 übertragen. So kann die Unified-Memory-Technologie einen einzigen, einheitlichen virtuellen Adressraum für den Speicher sowohl der CPU(s) 1706 als auch der GPU(s) 1708 ermöglichen und dadurch die Programmierung der GPU(s) 1708 und die Portierung von Anwendungen auf die GPU(s) 1708 vereinfachen.

[0160] Darüber hinaus können die GPU(s) 1708 einen Zugriffszähler enthalten, der die Häufigkeit des Zugriffs der GPU(s) 1708 auf den Speicher anderer Prozessoren verfolgen kann. Der Zugriffszähler kann dazu beitragen, dass Speicherseiten in den physikalischen Speicher desjenigen Prozessors verschoben werden, der am häufigsten auf die Seiten zugreift.

[0161] Der/die SoC(s) 1704 können eine beliebige Anzahl von Cache(s) 1712 enthalten, einschließlich der hier beschriebenen. Beispielsweise kann der/die

Cache(s) 1712 einen L3-Cache umfassen, der sowohl der/den CPU(s) 1706 als auch der/den GPU (s) 1708 zur Verfügung steht (z. B. der sowohl mit der/den CPU(s) 1706 als auch mit der/den GPU(s) 1708 verbunden ist). Der/die Cache(s) 1712 können einen Write-Back-Cache umfassen, der die Zustände von Zeilen verfolgen kann, z. B. durch Verwendung eines Cache-Kohärenzprotokolls (z. B. MEI, MESI, MSI usw.). Der L3-Cache kann je nach Ausführungsform 4 MB oder mehr umfassen, obwohl auch kleinere Cache-Größen verwendet werden können.

[0162] Der/die SoC(s) 1704 können eine arithmetische Logikeinheit(en) (ALU(s)) enthalten, die bei der Durchführung von Verarbeitungen in Bezug auf eine beliebige Vielzahl von Aufgaben oder Operationen des Fahrzeugs 1700 - wie die Verarbeitung von DNNs - genutzt werden kann. Darüber hinaus können der (die) SoC(s) 1704 eine Gleitkommaeinheit (en) (FPU(s)) - oder andere mathematische Coprozessoren oder numerische Coprozessortypen - zur Durchführung mathematischer Operationen innerhalb des Systems enthalten. Beispielsweise können die SoC(s) 104 eine oder mehrere FPUs enthalten, die als Ausführungseinheiten in einer oder mehreren CPU(s) 1706 und/oder GPU(s) 1708 integriert sind.

[0163] Der/die SoC(s) 1704 können einen oder mehrere Beschleuniger 1714 enthalten (z. B. Hardware-Beschleuniger, Software-Beschleuniger oder eine Kombination davon). Beispielsweise können die SoC(s) 1704 einen Hardware-Beschleunigungs-Cluster enthalten, der optimierte Hardware-Beschleuniger und/oder einen großen On-Chip-Speicher umfassen kann. Der große On-Chip-Speicher (z. B. 4 MB SRAM) kann den Hardware-Beschleunigungscluster in die Lage versetzen, neuronale Netzwerke und andere Berechnungen zu beschleunigen. Der Hardware-Beschleunigungscluster kann zur Ergänzung der GPU(s) 1708 und zur Entlastung einiger Aufgaben der GPU(s) 1708 verwendet werden (z. B. um mehr Zyklen der GPU(s) 1708 für die Durchführung anderer Aufgaben freizugeben). Der/die Beschleuniger 1714 können beispielsweise für gezielte Arbeitslasten (z. B. Wahrnehmung, Faltungsneuronale Netze (CNNs) usw.) verwendet werden, die stabil genug sind, um beschleunigt werden zu können. Der hier verwendete Begriff „CNN“ kann alle Arten von CNNs umfassen, einschließlich regionenbasierter oder regionaler neuronaler Faltungsnetze (RCNNs) und schneller RCNNs (z. B. zur Objekterkennung).

[0164] Der/die Beschleuniger 1714 (z. B. der Hardware-Beschleunigungscluster) können einen Deep-Learning-Beschleuniger (DLA) enthalten. Der/die DLA(s) können eine oder mehrere Tensor Processing Units (TPUs) umfassen, die so konfiguriert sein können, dass sie zusätzliche zehn Billionen Operationen pro Sekunde für Deep-Learning-Anwendungen und Inferencing bereitstellen. Bei den TPUs kann es sich um Beschleuniger handeln, die für die Ausführung von Bildverarbeitungsfunktionen (z. B. für CNNs, RCNNs usw.) konfiguriert und optimiert sind. Die DLA(s) können darüber hinaus für einen bestimmten Satz von neuronalen Netzwerktypen und Gleitkommaoperationen sowie für Inferenzierung optimiert sein. Das Design der DLA(s) kann mehr Leistung pro Millimeter bieten als eine Allzweck-GPU und übertrifft die Leistung einer CPU bei weitem. Die TPU(s) können mehrere Funktionen ausführen, einschließlich einer Einzelinstanz-Faltungsfunktion, die z. B. INT8-, INT16- und FP16-Datentypen sowohl für Merkmale als auch für Gewichte sowie Nachprozessorfunktionen unterstützt.

[0165] Die DLA(s) können schnell und effizient neuronale Netze, insbesondere CNNs, auf verarbeiteten oder unverarbeiteten Daten für eine Vielzahl von Funktionen ausführen, darunter beispielsweise und ohne Einschränkung: ein CNN für die Identifizierung und Erkennung von Objekten unter Verwendung von Daten von Kamerasensoren; ein CNN für die Abstandsschätzung unter Verwendung von Daten von Kamerasensoren; ein CNN für die Erkennung und Identifizierung von Einsatzfahrzeugen und die Erkennung unter Verwendung von Daten von Mikrofonen; ein CNN für die Gesichtserkennung und die Identifizierung von Fahrzeugbesitzern unter Verwendung von Daten von Kamerasensoren; und/oder ein CNN für sicherheitsrelevante Ereignisse.

[0166] Die DLA(s) können jede Funktion der GPU(s) 1708 ausführen, und durch die Verwendung eines Inferenzbeschleunigers kann ein Entwickler beispielsweise entweder die DLA(s) oder die GPU(s) 1708 für eine beliebige Funktion einsetzen. Zum Beispiel kann der Entwickler die Verarbeitung von CNNs und Gleitkommaoperationen auf die DLA(s) konzentrieren und andere Funktionen der GPU(s) 1708 und/oder anderen Beschleunigern 1714 überlassen.

[0167] Der/die Beschleuniger 1714 (z. B. der Hardware-Beschleunigungscluster) können einen programmierbaren Bildverarbeitungsbeschleuniger (PVA) umfassen, der hier alternativ auch als Computer-Vision-Beschleuniger bezeichnet werden kann. Der/die PVA(s) können zur Beschleunigung von Computer-Vision-Algorithmen für fortschrittliche Fahrerassistenzsysteme (ADAS), autonomes Fahren und/oder Augmented-Reality- (AR) und/oder Virtual-Reality- (VR) Anwendungen entwickelt und konfiguriert werden. Die PVA(s) können ein Gleichgewicht zwischen Leistung und Flexibilität bieten. So kann jede PVA zum Beispiel und ohne Einschränkung eine beliebige Anzahl von RISC-Kernen (Computer mit reduziertem Befehlssatz), direkten Speicherzu-

griff (DMA) und/oder eine beliebige Anzahl von Vektorprozessoren umfassen.

[0168] Die RISC-Kerne können mit Bildsensoren (z. B. den Bildsensoren einer der hier beschriebenen Kameras), Bildsignalprozessoren und/oder dergleichen zusammenwirken. Jeder der RISC-Kerne kann eine beliebige Menge an Speicher enthalten. Die RISC-Kerne können je nach Ausführungsform eine beliebige Anzahl von Protokollen verwenden. In einigen Beispielen können die RISC-Kerne ein Echtzeitbetriebssystem (RTOS) ausführen. Die RISC-Kerne können mit einem oder mehreren integrierten Schaltkreisen, anwendungsspezifischen integrierten Schaltkreisen (ASICs) und/oder Speicherbausteinen implementiert werden. Die RISC-Kerne können beispielsweise einen Befehls-Cache und/oder ein eng gekoppeltes RAM enthalten.

[0169] Der DMA kann es Komponenten der PVA(s) ermöglichen, unabhängig von der/den CPU(s) 1706 auf den Systemspeicher zuzugreifen. Der DMA kann eine beliebige Anzahl von Funktionen unterstützen, die zur Optimierung der PVA verwendet werden, einschließlich, aber nicht beschränkt auf die Unterstützung von mehrdimensionaler Adressierung und/oder zirkulärer Adressierung. In einigen Beispielen kann der DMA bis zu sechs oder mehr Dimensionen der Adressierung unterstützen, die Blockbreite, Blockhöhe, Blocktiefe, horizontale Blockabstufung, vertikale Blockabstufung und/oder Tiefenabstufung umfassen können.

[0170] Bei den Vektorprozessoren kann es sich um programmierbare Prozessoren handeln, die so konzipiert sein können, dass sie die Programmierung von Computer-Vision-Algorithmen effizient und flexibel ausführen und Signalverarbeitungsfunktionen bereitstellen. In einigen Beispielen kann die PVA einen PVA-Kern und zwei Vektorverarbeitungs-Subsystem-Partitionen umfassen. Der PVA-Kern kann ein Prozessor-Subsystem, DMA-Engine(s) (z. B. zwei DMA-Engines) und/oder andere Peripheriegeräte umfassen. Das Vektorverarbeitungs-Teilsystem kann als primäre Verarbeitungseinheit der PVA fungieren und eine Vektorverarbeitungseinheit (VPU), einen Befehlscache und/oder einen Vektorspeicher (z. B. VMEM) umfassen. Ein VPU-Kern kann einen digitalen Signalprozessor wie beispielsweise einen SIMD-Digitalprozessor (Single Instruction, Multiple Data) mit sehr langen Befehlsworten (VLIW) umfassen. Die Kombination von SIMD und VLIW kann den Durchsatz und die Geschwindigkeit erhöhen.

[0171] Jeder der Vektorprozessoren kann einen Befehls-Cache enthalten und mit einem dedizierten Speicher verbunden sein. Daher kann in einigen Beispielen jeder der Vektorprozessoren so konfiguriert sein, dass er unabhängig von den anderen Vektorprozessoren arbeitet. In anderen Beispielen können die Vektorprozessoren, die in einer bestimmten PVA enthalten sind, so konfiguriert sein, dass sie Datenparallelität verwenden. In einigen Ausführungsformen kann beispielsweise die Mehrzahl der in einer einzigen PVA enthaltenen Vektorprozessoren denselben Computer-Vision-Algorithmus ausführen, jedoch für unterschiedliche Bereiche eines Bildes. In anderen Beispielen können die in einer bestimmten PVA enthaltenen Vektorprozessoren gleichzeitig verschiedene Computer-Vision-Algorithmen für dasselbe Bild oder sogar verschiedene Algorithmen für aufeinander folgende Bilder oder Teile eines Bildes ausführen. Unter anderem kann eine beliebige Anzahl von PVAs im Hardware-Beschleunigungscluster enthalten sein und eine beliebige Anzahl von Vektorprozessoren in jeder der PVAs. Darüber hinaus können die PVA(s) einen zusätzlichen ECC-Speicher (Error Correcting Code) enthalten, um die Sicherheit des Gesamtsystems zu erhöhen.

[0172] Der/die Beschleuniger 1714 (z. B. der Hardware-Beschleunigungscluster) können ein Computer-Vision-Netzwerk auf dem Chip und SRAM enthalten, um ein SRAM mit hoher Bandbreite und niedriger Latenz für den/die Beschleuniger 1714 bereitzustellen. In einigen Beispielen kann der On-Chip-Speicher mindestens 4 MB SRAM umfassen, der beispielsweise und ohne Einschränkung aus acht feldkonfigurierbaren Speicherblöcken besteht, auf die sowohl die PVA als auch die DLA zugreifen können. Jedes Paar von Speicherblöcken kann eine APB-Schnittstelle (Advanced Peripheral Bus), Konfigurationsschaltungen, einen Controller und einen Multiplexer umfassen. Es kann jeder beliebige Speichertyp verwendet werden. Die PVA und die DLA können auf den Speicher über ein Backbone zugreifen, das der PVA und der DLA einen Hochgeschwindigkeitszugriff auf den Speicher ermöglicht. Das Backbone kann ein Computer-Vision-Netzwerk auf dem Chip umfassen, das die PVA und die DLA mit dem Speicher verbindet (z. B. unter Verwendung des APB).

[0173] Das Computer-Vision-Netz auf dem Chip kann eine Schnittstelle enthalten, die vor der Übertragung von Steuersignalen/Adressen/Daten feststellt, dass sowohl die PVA als auch die DLA einsatzbereite und gültige Signale liefern. Eine solche Schnittstelle kann getrennte Phasen und getrennte Kanäle für die Übertragung von Steuersignalen/Adressen/Daten sowie eine Burst-Kommunikation für die kontinuierliche Datenübertragung vorsehen. Diese Art von Schnittstelle kann den Normen ISO 26262 oder IEC 61508 entsprechen, es können aber auch andere Normen und Protokolle verwendet werden.

[0174] In einigen Beispielen können die SoC(s) 1704 einen Echtzeit-Raytracing-Hardwarebeschleuniger enthalten, wie er in der US-Patentanmeldung

Nr. 16/101,232, eingereicht am 10. August 2018, beschrieben ist. Der Echtzeit-Raytracing-Hardwarebeschleuniger kann zur schnellen und effizienten Bestimmung der Positionen und Ausdehnungen von Objekten (z. B. innerhalb eines Weltmodells), zur Erzeugung von Echtzeit-Visualisierungssimulationen, zur RADAR-Signalinterpretation, zur Schallausbreitungssynthese und/oder -analyse, zur Simulation von SONAR-Systemen, zur allgemeinen Wellenausbreitungssimulation, zum Vergleich mit LIDAR-Daten zum Zwecke der Lokalisierung und/oder anderer Funktionen und/oder für andere Zwecke verwendet werden. In einigen Ausführungsformen können eine oder mehrere Tree Traversal Units (TTUs) für die Ausführung einer oder mehrerer Operationen im Zusammenhang mit der Strahlenverfolgung verwendet werden.

[0175] Der/die Beschleuniger 1714 (z. B. der Hardware-Beschleuniger-Cluster) können für das autonome Fahren auf vielfältige Weise eingesetzt werden. Der PVA kann ein programmierbarer Bildverarbeitungsbeschleuniger sein, der für wichtige Verarbeitungsschritte in ADAS und autonomen Fahrzeugen verwendet werden kann. Die Fähigkeiten der PVA eignen sich gut für algorithmische Bereiche, die eine vorhersehbare Verarbeitung bei geringem Stromverbrauch und geringer Latenzzeit erfordern. Mit anderen Worten: Die PVA eignet sich gut für halbdichte oder dichte reguläre Berechnungen, selbst bei kleinen Datensätzen, die vorhersehbare Laufzeiten mit geringer Latenz und geringem Stromverbrauch erfordern. Im Zusammenhang mit Plattformen für autonome Fahrzeuge sind die PVAs daher für die Ausführung klassischer Computer-Vision-Algorithmen konzipiert, da sie bei der Objekterkennung und der Verarbeitung ganzzahliger mathematischer Daten effizient sind.

[0176] Bei einer Ausführungsform der Technologie wird die PVA beispielsweise für die Durchführung des Computer-Stereobildes verwendet. In einigen Beispielen kann ein auf semiglobaler Anpassung basierender Algorithmus verwendet werden, obwohl dies nicht als Einschränkung gedacht ist. Viele Anwendungen für das autonome Fahren der Stufen 3 bis 5 erfordern eine Bewegungsabschätzung/Stereoabgleich während der Fahrt (z. B. Struktur aus Bewegung, Fußgängererkennung, Fahrspurerkennung usw.). Die PVA kann eine Computer-Stereosichtfunktion mit Eingaben von zwei monokularen Kameras durchführen.

[0177] In einigen Beispielen kann die PVA verwendet werden, um einen dichten optischen Fluss durchzuführen. Entsprechend der Verarbeitung von RADAR-Rohdaten (z. B. unter Verwendung einer 4D-Fast-Fourier-Transformation), um verarbeitete RADAR-Daten zu erhalten. In anderen Beispielen wird die PVA für die Verarbeitung von Flugzeittiefendaten verwendet, z. B. durch Verarbeitung von Flugzeit-Rohdaten, um verarbeitete Flugzeitdaten zu erhalten.

[0178] Mit dem DLA kann jede Art von Netzwerk betrieben werden, um die Kontrolle und die Fahrsicherheit zu verbessern, z. B. ein neuronales Netzwerk, das für jede Objekterkennung einen Konfidenzwert ausgibt. Ein solcher Konfidenzwert kann als Wahrscheinlichkeit oder als relative „Gewichtung“ der einzelnen Erkennungen im Vergleich zu anderen Erkennungen interpretiert werden. Dieser Konfidenzwert ermöglicht es dem System, weitere Entscheidungen darüber zu treffen, welche Erkennungen als echte positive Erkennungen und welche als falschpositive Erkennungen betrachtet werden sollten. So kann das System beispielsweise einen Schwellenwert für die Konfidenz festlegen und nur die Erkennungen, die diesen Schwellenwert überschreiten, als echte positive Erkennungen betrachten. In einem automatischen Notbremssystem (AEB) würden falsch positive Erkennungen dazu führen, dass das Fahrzeug automatisch eine Notbremsung durchführt, was natürlich unerwünscht ist. Daher sollten nur die sichersten Erkennungen als Auslöser für AEB in Betracht gezogen werden. Die DLA kann ein neuronales Netz zur Regression des Konfidenzwertes einsetzen. Das neuronale Netz kann als Eingabe mindestens eine Teilmenge von Parametern verwenden, wie z. B. die Abmessungen des Begrenzungsrahmens, die (z. B. von einem anderen Teilsystem) erhaltene Schätzung der Bodenebene, die Ausgabe des IMU-Sensors 1766, die mit der Ausrichtung des Fahrzeugs 1700 korreliert, die Entfernung, die Schätzungen der 3D-Position des Objekts, die vom neuronalen Netz und/oder anderen Sensoren (z. B. LIDAR-Sensor(en) 1764 oder RADAR-Sensor(en) 1760) erhalten werden, und andere.

[0179] Der/die SoC(s) 1704 können Datenspeicher 1716 (z. B. Speicher) enthalten. Der (die) Datenspeicher 1716 können ein On-Chip-Speicher des (der) SoC(s) 1704 sein, der (die) neuronale(n) Netzwerke speichern können, die auf der GPU und/oder der DLA ausgeführt werden. In einigen Beispielen kann die Kapazität des/der Datenspeicher(s) 1716 groß genug sein, um mehrere Instanzen von neuronalen Netzen aus Gründen der Redundanz und Sicherheit zu speichern. Der/die Datenspeicher 1712 können L2 oder L3 Cache(s) 1712 umfassen. Der Verweis auf den/die Datenspeicher 1716 kann einen Verweis auf den Speicher beinhalten, der mit der PVA, DLA und/oder anderen Beschleunigern 1714 verbunden ist, wie hier beschrieben.

[0180] Der/die SoC(s) 1704 können einen oder mehrere Prozessor(en) 1710 (z. B. eingebettete Prozessoren) enthalten. Der (die) Prozessor(en) 1710 können einen Boot- und Energieverwaltungsprozessor umfassen, der ein dedizierter Prozessor und ein

Subsystem sein kann, um Boot-Energie- und -Verwaltungsfunktionen und die damit verbundene Sicherheitsdurchsetzung zu handhaben. Der Boot- und Energieverwaltungsprozessor kann Teil der Bootsequenz des/der SoC(s) 1704 sein und kann Laufzeit-Energieverwaltungsdienste bereitstellen. Der Boot-Energieversorgungs- und -Verwaltungsprozessor kann Takt- und Spannungsprogrammierung, Unterstützung bei Systemübergängen in einen Zustand mit niedriger Leistung, Verwaltung von SoC(s) 1704-Temperaturen und Temperatursensoren und/oder Verwaltung der SoC(s) 1704-Energieversorgungszustände bieten. Jeder Temperatursensor kann als Ringoszillator implementiert sein, dessen Ausgangsfrequenz proportional zur Temperatur ist, und der/die SoC(s) 1704 können die Ringoszillatoren verwenden, um die Temperaturen der CPU(s) 1706, der GPU(s) 1708 und/oder des/der Beschleuniger(s) 1714 zu erfassen. Wenn festgestellt wird, dass die Temperaturen einen Schwellenwert überschreiten, kann der Boot- und Energieverwaltungsprozessor in eine Temperaturfehlerroutine eintreten und den/die SoC(s) 1704 in einen Zustand mit geringerer Leistung versetzen und/oder das Fahrzeug 1700 in einen Chauffeur-zu-sicherem-Halt-Modus versetzen (z. B. das Fahrzeug 1700 zu einem sicheren Halt bringen).

[0181] Der/die Prozessor(en) 1710 können außerdem eine Reihe eingebetteter Prozessoren enthalten, die als Audioverarbeitungsmodul dienen können. Bei der Audioverarbeitungs-Engine kann es sich um ein Audio-Subsystem handeln, das eine vollständige Hardware-Unterstützung für Mehrkanal-Audio über mehrere Schnittstellen sowie eine breite und flexible Palette von Audio-I/O-Schnittstellen ermöglicht. In einigen Beispielen ist die Audioverarbeitungs-Engine ein dedizierter Prozessorkern mit einem digitalen Signalprozessor mit dediziertem RAM.

[0182] Der (die) Prozessor(en) 1710 können außerdem eine ständig eingeschaltete Prozessor-Engine enthalten, die die notwendigen Hardware-Funktionen zur Unterstützung der Sensorverwaltung mit geringem Stromverbrauch und des Aufwachens von Anwendungsfällen bereitstellen kann. Die „always on“-Prozessor-Engine kann einen Prozessorkern, einen eng gekoppelten Arbeitsspeicher, unterstützende Peripheriegeräte (z. B. Zeitgeber und Interrupt-Controller), verschiedene I/O-Controller-Peripheriegeräte und Routing-Logik umfassen.

[0183] Der (die) Prozessor(en) 1710 können außerdem eine Sicherheits-Cluster-Engine enthalten, die ein spezielles Prozessor-Subsystem für das Sicherheitsmanagement von Kfz-Anwendungen umfasst. Die Sicherheits-Cluster-Engine kann zwei oder mehr Prozessorkerne, einen eng gekoppelten Arbeitsspeicher, unterstützende Peripheriegeräte (z. B. Zeitgeber, einen Interrupt-Controller usw.) und/oder eine Routing-Logik umfassen. In einem Sicherheitsmodus können die zwei oder mehr Kerne in einem Lockstep-Modus arbeiten und als ein einziger Kern mit einer Vergleichslogik zur Erkennung von Unterschieden zwischen ihren Operationen fungieren.

[0184] Der (die) Prozessor(en) 1710 können außerdem eine Echtzeit-Kamera-Engine enthalten, die ein spezielles Prozessor-Subsystem für die Verwaltung der Echtzeit-Kamera umfassen kann.

[0185] Der (die) Prozessor(en) 1710 können außerdem einen Signalprozessor mit hohem Dynamikbereich umfassen, der einen Bildsignalprozessor enthalten kann, der eine Hardware-Engine ist, die Teil der Kameraverarbeitungspipeline ist.

[0186] Der/die Prozessor(en) 1710 können einen Videobild-Compositor enthalten, der ein Verarbeitungsblock sein kann (z. B. auf einem Mikroprozessor implementiert), der Videonachbearbeitungsfunktionen implementiert, die von einer Videowiedergabeanwendung benötigt werden, um das endgültige Bild für das Player-Fenster zu erzeugen. Der Videobild-Compositor kann eine Linsenverzerrungskorrektur an der (den) Weitwinkelkamera(s) 1770, an der (den) Surround-Kamera(s) 1774 und/oder an den Kamerasensoren der Kabinenüberwachung vornehmen. Der kabineninterne Überwachungskamerasensor wird vorzugsweise von einem neuronalen Netz überwacht, das auf einer anderen Instanz des Advanced SoC läuft und so konfiguriert ist, dass es Ereignisse in der Kabine erkennt und entsprechend reagiert. Ein System in der Kabine kann Lippenlesen durchführen, um den Mobilfunkdienst zu aktivieren und einen Anruf zu tätigen, E-Mails zu diktieren, den Zielort des Fahrzeugs zu ändern, das Infotainmentsystem und die Einstellungen des Fahrzeugs zu aktivieren oder zu ändern oder sprachgesteuertes Surfen im Internet zu ermöglichen. Bestimmte Funktionen stehen dem Fahrer nur zur Verfügung, wenn das Fahrzeug in einem autonomen Modus betrieben wird, und sind ansonsten deaktiviert.

[0187] Der Videobild-Compositor kann eine verbesserte zeitliche Rauschunterdrückung sowohl zur räumlichen als auch zur zeitlichen Rauschunterdrückung enthalten. Tritt beispielsweise Bewegung in einem Video auf, gewichtet die Rauschunterdrückung die räumlichen Informationen entsprechend und verringert das Gewicht der Informationen, die von benachbarten Bildern stammen. Wenn ein Bild oder ein Teil eines Bildes keine Bewegung enthält, kann die vom Video-Compositor durchgeführte zeitliche Rauschunterdrückung Informationen aus dem vorherigen Bild verwenden, um das Rauschen im aktuellen Bild zu reduzieren.

[0188] Der Videobild-Compositor kann auch so konfiguriert sein, dass er eine Stereobildentzerrung an eingegebenen Stereoobjektivframes durchführt. Der Videobild-Compositor kann außerdem für die Gestaltung der Benutzeroberfläche verwendet werden, wenn der Desktop des Betriebssystems in Gebrauch ist und die GPU(s) 1708 nicht ständig neue Oberflächen rendern muss. Selbst wenn die GPUs 1708 eingeschaltet und aktiv mit dem 3D-Rendering beschäftigt ist/sind, kann der Videobild-Compositor verwendet werden, um die GPUs 1708 zu entlasten und die Leistung und Reaktionsfähigkeit zu verbessern.

[0189] Der/die SoC(s) 1704 können außerdem eine serielle MIPI-Kameraschnittstelle für den Empfang von Video und Eingaben von Kameras, eine Hochgeschwindigkeitsschnittstelle und/oder einen Videoeingabeblock enthalten, der für Kamera- und verwandte Pixeleingabefunktionen verwendet werden kann. Der/die SoC(s) 1704 können außerdem einen Eingangs-/Ausgangs-Controller enthalten, der/die durch Software gesteuert werden können und für den Empfang von I/O-Signalen verwendet werden können, die nicht an eine bestimmte Rolle gebunden sind.

[0190] Der/die SoC(s) 1704 können außerdem eine breite Palette von Peripherieschnittstellen umfassen, um die Kommunikation mit Peripheriegeräten, Audiocodecs, Energieverwaltung und/oder anderen Geräten zu ermöglichen. Der/die SoC(s) 1704 können verwendet werden, um Daten von Kameras (z. B. verbunden über Gigabit Multimedia Serial Link und Ethernet), Sensoren (z. B. LIDAR-Sensor(en) 1764, RADAR-Sensor(en) 1760 usw., die über Ethernet verbunden sein können), Daten vom Bus 1702 (z. B. Geschwindigkeit des Fahrzeugs 1700, Lenkradposition usw.), Daten von GNSS-Sensor(en) 1758 (z. B. verbunden über Ethernet oder CAN-Bus) zu verarbeiten. Der/die SoC(s) 1704 können darüber hinaus dedizierte Hochleistungs-Massenspeicher-Controller enthalten, die ihre eigenen DMA-Engines enthalten können und die dazu verwendet werden können, die CPU(s) 1706 von routinemäßigen Datenverwaltungsaufgaben zu entlasten.

[0191] Der/die SoC(s) 1704 können eine End-to-End-Plattform mit einer flexiblen Architektur sein, die die Automatisierungsebenen 3 bis 5 abdeckt und dadurch eine umfassende funktionale Sicherheitsarchitektur bietet, die Computer-Vision- und ADAS-Techniken für Diversität und Redundanz nutzt und eine Plattform für einen flexiblen, zuverlässigen Fahrsoftware-Stack zusammen mit Deep-Learning-Tools bereitstellt. Die SoC(s) 1704 können schneller, zuverlässiger und sogar energie- und platzsparender sein als herkömmliche Systeme. Zum Beispiel kann der/die Beschleuniger 1714 in Kombination mit der/den CPU(s) 1706, der/den GPU(s) 1708 und dem/den Datenspeicher(n) 1716 eine schnelle, effiziente Plattform für autonome Fahrzeuge der Stufe 3-5 bilden.

[0192] Die Technologie bietet somit Fähigkeiten und Funktionen, die mit herkömmlichen Systemen nicht erreicht werden können. Beispielsweise können Computer-Vision-Algorithmen auf CPUs ausgeführt werden, die mit Hilfe von Hochsprachen wie der Programmiersprache C so konfiguriert werden können, dass sie eine Vielzahl von Verarbeitungsalgorithmen für eine Vielzahl von visuellen Daten ausführen können. Allerdings sind CPUs oft nicht in der Lage, die Leistungsanforderungen vieler Bildverarbeitungsanwendungen zu erfüllen, z. B. in Bezug auf die Ausführungszeit und den Stromverbrauch. Insbesondere sind viele CPUs nicht in der Lage, komplexe Objekterkennungsalgorithmen in Echtzeit auszuführen, was eine Voraussetzung für fahrzeuginterne ADAS-Anwendungen und eine Voraussetzung für praktische autonome Fahrzeuge der Stufe 3-5 ist.

[0193] Im Gegensatz zu herkömmlichen Systemen ermöglicht die hier beschriebene Technologie durch die Bereitstellung eines CPU-Komplexes, eines GPU-Komplexes und eines Hardware-Beschleunigungs-Clusters, dass mehrere neuronale Netze gleichzeitig und/oder nacheinander ausgeführt und die Ergebnisse miteinander kombiniert werden können, um autonome Fahrfunktionen der Stufe 3-5 zu ermöglichen. Beispielsweise kann ein CNN, das auf der DLA oder dGPU (z. B. die GPU(s) 1720) ausgeführt wird, eine Text- und Worterkennung umfassen, die es dem Supercomputer ermöglicht, Verkehrsschilder zu lesen und zu verstehen, einschließlich Schildern, für die das neuronale Netz nicht speziell trainiert wurde. Die DLA kann ferner ein neuronales Netz enthalten, das in der Lage ist, das Zeichen zu identifizieren, zu interpretieren und semantisch zu verstehen und dieses semantische Verständnis an die auf dem CPU-Komplex laufenden Wegplanungsmodule weiterzugeben.

[0194] Ein weiteres Beispiel ist, dass mehrere neuronale Netze gleichzeitig betrieben werden können, wie es für das Fahren der Stufen 3, 4 oder 5 erforderlich ist. So kann beispielsweise ein Warnschild mit der Aufschrift „Vorsicht: Blinkende Lichter weisen auf Glatteis hin“ zusammen mit einem elektrischen Licht von mehreren neuronalen Netzen unabhängig oder gemeinsam interpretiert werden. Das Schild selbst kann von einem ersten eingesetzten neuronalen Netz (z. B. einem trainierten neuronalen Netz) als Verkehrsschild identifiziert werden, der Text „Blinkende Lichter deuten auf Glatteis hin“ kann von einem zweiten eingesetzten neuronalen Netz interpretiert werden, das die Wegplanungssoftware des Fahrzeugs (die vorzugsweise auf dem CPU-Komplex ausgeführt wird) darüber informiert, dass beim Erkennen blinkender Lichter Glatteis vorliegt. Das

Blinklicht kann durch den Betrieb eines dritten neuronalen Netzes über mehrere Frames hinweg identifiziert werden, das die Wegplanungssoftware des Fahrzeugs über das Vorhandensein (oder Fehlen) von Blinklichtern informiert. Alle drei neuronalen Netze können gleichzeitig laufen, z. B. innerhalb der DLA und/oder auf der/den GPU(s) 1708.

[0195] In einigen Beispielen kann ein CNN zur Gesichtserkennung und Identifizierung des Fahrzeugbesitzers Daten von Kamerasensoren verwenden, um die Anwesenheit eines autorisierten Fahrers und/oder Besitzers des Fahrzeugs 1700 zu erkennen. Die „always on"-Sensorverarbeitungs-Engine kann verwendet werden, um das Fahrzeug zu entriegeln, wenn der Besitzer sich der Fahrertür nähert und die Lichter einschaltet, und um im Sicherheitsmodus das Fahrzeug zu deaktivieren, wenn der Besitzer das Fahrzeug verlässt. Auf diese Weise sorgen die SoC(s) 1704 für Sicherheit gegen Diebstahl und/oder Carjacking.

[0196] In einem anderen Beispiel kann ein CNN zur Erkennung und Identifizierung von Einsatzfahrzeugen Daten von Mikrofonen 1796 verwenden, um Sirenen von Einsatzfahrzeugen zu erkennen und zu identifizieren. Im Gegensatz zu herkömmlichen Systemen, die allgemeine Klassifikatoren zur Erkennung von Sirenen und zur manuellen Extraktion von Merkmalen verwenden, nutzen die SoC(s) 1704 das CNN zur Klassifizierung von Umgebungs- und Stadtgeräuschen sowie zur Klassifizierung visueller Daten. In einer bevorzugten Ausführungsform wird der CNN, der auf dem DLA läuft, darauf trainiert, die relative Annäherungsgeschwindigkeit des Einsatzfahrzeugs zu erkennen (z. B. durch Verwendung des Dopplereffekts). Das CNN kann auch so trainiert werden, dass es Einsatzfahrzeuge erkennt, die spezifisch für den lokalen Bereich sind, in dem das Fahrzeug unterwegs ist, wie von GNSS-Sensor(en) 1758 ermittelt. So wird das CNN z. B. bei einem Einsatz in Europa versuchen, europäische Sirenen zu erkennen, und bei einem Einsatz in den Vereinigten Staaten wird das CNN versuchen, nur nordamerikanische Sirenen zu erkennen. Sobald ein Einsatzfahrzeug erkannt wird, kann ein Steuerprogramm verwendet werden, um eine Sicherheitsroutine für Einsatzfahrzeuge auszuführen, das Fahrzeug zu verlangsamen, an den Straßenrand zu fahren, das Fahrzeug zu parken und/oder das Fahrzeug im Leerlauf laufen zu lassen, mit Hilfe der Ultraschallsensoren 1762, bis das/die Einsatzfahrzeug(e) vorbeifahren.

[0197] Das Fahrzeug kann eine oder mehrere CPU(s) 1718 (z. B. diskrete CPU(s) oder dCPU(s)) enthalten, die über eine Hochgeschwindigkeitsverbindung (z. B. PCIe) mit dem/den SoC(s) 1704 verbunden sein können. Die CPU(s) 1718 können beispielsweise einen X86-Prozessor umfassen. Die CPU(s) 1718 können verwendet werden, um eine Vielzahl von Funktionen auszuführen, einschließlich der Schlichtung potenziell inkonsistenter Ergebnisse zwischen ADAS-Sensoren und dem/den SoC(s) 1704 und/oder der Überwachung des Status und des Zustands des/der Controllers) 1736 und/oder des Infotainment-SoC 1730, zum Beispiel.

[0198] Das Fahrzeug 1700 kann eine oder mehrere GPU(s) 1720 (z. B. diskrete GPU(s) oder dGPU(s)) enthalten, die über eine Hochgeschwindigkeitsverbindung (z. B. NVIDIAs NVLINK) mit dem/den SoC(s) 1704 gekoppelt sein können. Der/die GPU(s) 1720 können zusätzliche Funktionen der künstlichen Intelligenz bereitstellen, z. B. durch die Ausführung redundanter und/oder unterschiedlicher neuronaler Netze, und können zum Trainieren und/oder Aktualisieren neuronaler Netze basierend auf Eingaben (z. B. Sensordaten) von Sensoren des Fahrzeugs 1700 verwendet werden.

[0199] Das Fahrzeug 1700 kann ferner die Netzwerkschnittstelle 1724 enthalten, die eine oder mehrere drahtlose Antennen 1726 (z. B. eine oder mehrere drahtlose Antennen für verschiedene Kommunikationsprotokolle, wie eine Mobilfunkantenne, eine Bluetooth-Antenne usw.) enthalten kann. Die Netzwerkschnittstelle 1724 kann verwendet werden, um eine drahtlose Verbindung über das Internet mit der Cloud (z. B. mit dem/den Server(n) 1778 und/oder anderen Netzwerkgeräten), mit anderen Fahrzeugen und/oder mit Computergeräten (z. B. Client-Geräten von Fahrgästen) zu ermöglichen. Zur Kommunikation mit anderen Fahrzeugen kann eine direkte Verbindung zwischen den beiden Fahrzeugen und/oder eine indirekte Verbindung hergestellt werden (z. B. über Netzwerke und das Internet). Direkte Verbindungen können über eine Fahrzeug-zu-Fahrzeug-Kommunikationsverbindung hergestellt werden. Die Fahrzeug-zu-Fahrzeug-Kommunikationsverbindung kann dem Fahrzeug 1700 Informationen über Fahrzeuge in der Nähe des Fahrzeugs 1700 liefern (z. B. Fahrzeuge vor, neben und/oder hinter dem Fahrzeug 1700). Diese Funktion kann Teil einer kooperativen adaptiven Geschwindigkeitsregelungsfunktion des Fahrzeugs 1700 sein.

[0200] Die Netzwerkschnittstelle 1724 kann einen SoC enthalten, der Modulations- und Demodulationsfunktionen bereitstellt und den/die Controller 1736 in die Lage versetzt, über drahtlose Netzwerke zu kommunizieren. Die Netzwerkschnittstelle 1724 kann ein Funkfrequenz-Frontend für die Aufwärtskonvertierung von Basisband auf Funkfrequenz und die Abwärtskonvertierung von Funkfrequenz auf Basisband enthalten. Die Frequenzumwandlungen können mit bekannten Verfahren und/oder mit Superheterodyn-Verfahren durchgeführt werden. In einigen Beispielen kann die Funkfrequenz-Front-End-Funktionalität durch einen separaten Chip bereitgestellt werden. Die Netzwerkschnittstelle

kann drahtlose Funktionen für die Kommunikation über LTE, WCDMA, UMTS, GSM, CDMA2000, Bluetooth, Bluetooth LE, Wi-Fi, Z-Wave, ZigBee, LoRaWAN und/oder andere drahtlose Protokolle enthalten.

[0201] Das Fahrzeug 1700 kann ferner einen oder mehrere Datenspeicher 1728 umfassen, die außerhalb des Chips (z. B. außerhalb der SoC(s) 1704) gespeichert werden können. Der/die Datenspeicher 1728 können ein oder mehrere Speicherelemente wie RAM, SRAM, DRAM, VRAM, Flash, Festplatten und/oder andere Komponenten und/oder Geräte umfassen, die mindestens ein Datenbit speichern können.

[0202] Das Fahrzeug 1700 kann außerdem GNSS-Sensor(en) 1758 enthalten. Der/die GNSS-Sensor (en) 1758 (z. B. GPS, unterstützte GPS-Sensoren, Differential-GPS-Sensoren (DGPS) usw.) unterstützen bei der Kartierung, Wahrnehmung, Erstellung von Belegungsrastern und/oder Wegplanungsfunktionen. Es kann eine beliebige Anzahl von GNSS-Sensoren 1758 verwendet werden, z. B. und ohne Einschränkung ein GPS, das einen USB-Anschluss mit einer Ethernet-zu-Seriell-Brücke (RS-232) verwendet.

[0203] Das Fahrzeug 1700 kann außerdem RADAR-Sensor(en) 1760 enthalten. Der/die RADAR-Sensor(en) 1760 können vom Fahrzeug 1700 zur Fahrzeugerkennung über große Entfernungen verwendet werden, auch bei Dunkelheit und/oder schlechten Wetterbedingungen. Der/die RADAR-Sensor(en) 1760 können den CAN-Bus und/oder den Bus 1702 (z. B. zur Übertragung von Daten, die von dem/den RADAR-Sensor(en) 1760 erzeugt wurden) zur Steuerung und zum Zugriff auf Objektverfolgungsdaten verwenden, wobei in einigen Beispielen der Zugriff auf Rohdaten über Ethernet erfolgt. Es kann eine Vielzahl von RADAR-Sensortypen verwendet werden. Beispielsweise und ohne Einschränkung können der/die RADAR-Sensor(en) 1760 für den Einsatz von Front-, Heck- und Seiten-RADAR geeignet sein. In einigen Beispielen werden Puls-Doppler-RADAR-Sensoren verwendet.

[0204] Der/die RADAR-Sensor(en) 1760 können verschiedene Konfigurationen umfassen, wie z. B. große Reichweite mit engem Sichtfeld, kurze Reichweite mit breitem Sichtfeld, seitliche Abdeckung kurzer Reichweite usw. In einigen Beispielen kann RADAR mit großer Reichweite für die adaptive Geschwindigkeitsregelung verwendet werden. Die RADAR-Systeme mit großer Reichweite können ein breites Sichtfeld bieten, das durch zwei oder mehr unabhängige Abtastungen, z. B. innerhalb einer Reichweite von 250 m, realisiert wird. Der/die RADAR-Sensor(en) 1760 können helfen, zwischen statischen und sich bewegenden Objekten zu unterscheiden, und können von ADAS-Systemen für Notbremsassistenten und Vorwärtskollisionswarnungen verwendet werden. Langstrecken-RADAR-Sensoren können monostatische multimodale RADAR mit mehreren (z. B. sechs oder mehr) festen RADAR-Antennen und einer Hochgeschwindigkeits-CAN- und FlexRay-Schnittstelle umfassen. In einem Beispiel mit sechs Antennen können die mittleren vier Antennen ein fokussiertes Strahlenmuster erzeugen, das dazu dient, die Umgebung des Fahrzeugs bei höheren Geschwindigkeiten mit minimalen Störungen durch den Verkehr auf den angrenzenden Fahrspuren zu erfassen. Die beiden anderen Antennen können das Sichtfeld erweitern, so dass Fahrzeuge, die in die 1700-Fahrspur des Fahrzeugs einfahren oder diese verlassen, schnell erkannt werden können.

[0205] RADAR-Systeme mit mittlerer Reichweite können beispielsweise eine Reichweite von bis zu 1760 m (vorne) oder 80 m (hinten) und ein Sichtfeld von bis zu 42 Grad (vorne) oder 1750 Grad (hinten) haben. Zu den RADAR-Systemen mit geringer Reichweite können unter anderem RADAR-Sensoren gehören, die an beiden Enden des hinteren Stoßfängers angebracht werden. Wenn ein solches RADAR-Sensorsystem an beiden Enden des hinteren Stoßfängers angebracht ist, kann es zwei Strahlen erzeugen, die den toten Winkel hinter und neben dem Fahrzeug ständig überwachen.

[0206] RADAR-Systeme mit geringer Reichweite können in einem ADAS-System zur Erkennung des toten Winkels und/oder als Spurwechselassistent eingesetzt werden.

[0207] Das Fahrzeug 1700 kann außerdem Ultraschallsensor(en) 1762 enthalten. Der/die Ultraschallsensor(en) 1762, der/die an der Vorderseite, an der Rückseite und/oder an den Seiten des Fahrzeugs 1700 angebracht sein können, können zur Einparkhilfe und/oder zur Erstellung und Aktualisierung eines Belegungsrasters verwendet werden. Es kann eine Vielzahl von Ultraschallsensoren 1762 verwendet werden, und unterschiedliche Ultraschallsensoren 1762 können für unterschiedliche Erfassungsbereiche (z. B. 2,5 m, 4 m) eingesetzt werden. Der/die Ultraschallsensor(en) 1762 können bei funktionalen Sicherheitsstufen von ASIL B arbeiten.

[0208] Das Fahrzeug 1700 kann LIDAR-Sensor(en) 1764 enthalten. Der/die LIDAR-Sensor(en) 1764 können für Objekt- und Fußgängererkennung, Notbremsung, Kollisionsvermeidung und/oder andere Funktionen verwendet werden. Der/die LIDAR-Sensor(en) 1764 können der funktionalen Sicherheitsstufe ASIL B entsprechen. In einigen Beispielen kann das Fahrzeug 1700 mehrere LIDAR-Sensoren 1764 (z. B. zwei, vier, sechs usw.) enthalten, die

Ethernet verwenden können (z. B. um Daten an einen Gigabit-Ethernet-Switch zu liefern).

[0209] In einigen Beispielen kann der/die LIDAR-Sensor(en) 1764 in der Lage sein, eine Liste von Objekten und deren Entfernungen für ein 360-Grad-Sichtfeld zu liefern. Im Handel erhältliche LIDAR-Sensoren 1764 können eine Reichweite von etwa 1700 m haben, mit einer Genauigkeit von 2 cm bis 3 cm und mit Unterstützung für eine Ethernet-Verbindung mit 1700 Mbit/s, zum Beispiel. In einigen Beispielen können ein oder mehrere nicht vorspringende LIDAR-Sensoren 1764 verwendet werden. In solchen Beispielen können der/die LIDAR-Sensor (en) 1764 als kleines Gerät implementiert werden, das in die Front, das Heck, die Seiten und/oder die Ecken des Fahrzeugs 1700 eingebettet werden kann. Der/die LIDAR-Sensor(en) 1764 können in solchen Beispielen ein horizontales Sichtfeld von bis zu 120 Grad und ein vertikales Sichtfeld von bis zu 35 Grad mit einer Reichweite von 200 m selbst bei Objekten mit geringem Reflektionsvermögen bieten. Der/die frontmontierte(n) LIDAR-Sensor(en) 1764 können für ein horizontales Sichtfeld zwischen 45 Grad und 135 Grad konfiguriert werden.

[0210] In einigen Beispielen können auch LIDAR-Technologien wie 3D-Blitz-LIDAR eingesetzt werden. 3D-Blitz-LIDAR verwendet einen Laserblitz als Übertragungsquelle, um die Umgebung des Fahrzeugs bis zu einer Entfernung von etwa 200 m zu beleuchten. Eine Flash-LIDAR-Einheit umfasst einen Empfänger, der die Laufzeit des Laserpulses und das reflektierte Licht auf jedem Pixel aufzeichnet, was wiederum der Entfernung zwischen dem Fahrzeug und den Objekten entspricht. Mit Flash-LIDAR lassen sich mit jedem Laserblitz hochpräzise und verzerrungsfreie Bilder der Umgebung erzeugen. In einigen Beispielen können vier Flash-LIDAR-Sensoren eingesetzt werden, einer auf jeder Seite des Fahrzeugs 1700. Zu den verfügbaren 3D-Blitz-LIDAR-Systemen gehört eine Festkörper-3D-Star-Array-LIDAR-Kamera, die außer einem Gebläse keine beweglichen Teile aufweist (z. B. ein nicht scannendes LIDAR-Gerät). Das Flash-LIDAR-Gerät kann einen 5-Nanosekunden-Laserimpuls der Klasse I (augensicher) pro Bild verwenden und das reflektierte Laserlicht in Form von 3D-Entfernungspunktwolken und koregistrierten Intensitätsdaten erfassen. Durch die Verwendung von Flash-LIDAR und weil Flash-LIDAR ein Festkörpergerät ohne bewegliche Teile ist, kann der/die LIDAR-Sensor (en) 1764 weniger anfällig für Bewegungsunschärfe, Vibrationen und/oder Stöße sein.

[0211] Das Fahrzeug kann außerdem IMU-Sensor (en) 1766 enthalten. Der/die IMU-Sensor(en) 1766 können in einigen Beispielen in der Mitte der Hinterachse des Fahrzeugs 1700 angeordnet sein. Der (die) IMU-Sensor(en) 1766 können beispielsweise und ohne Einschränkung einen (mehrere) Beschleunigungsmesser, einen (mehrere) Magnetometer, ein (mehrere) Gyroskop(e), einen (mehrere) Magnetkompass(e) und/oder andere Sensortypen umfassen. In einigen Beispielen, wie z. B. bei sechsachsigen Anwendungen, können der/die IMU-Sensor(en) 1766 Beschleunigungsmesser und Gyroskope umfassen, während bei neunachsigen Anwendungen der/die IMU-Sensor(en) 1766 Beschleunigungsmesser, Gyroskope und Magnetometer umfassen können.

[0212] In einigen Ausführungsformen kann der/die IMU-Sensor(en) 1766 als ein miniaturisiertes, hochleistungsfähiges GPS-gestütztes Trägheitsnavigationssystem (GPS/INS) implementiert werden, das mikroelektromechanische Trägheitssensoren (MEMS), einen hochempfindlichen GPS-Empfänger und fortschrittliche Kalman-Filteralgorithmen kombiniert, um Schätzungen von Position, Geschwindigkeit und Lage zu liefern. So kann in einigen Beispielen der IMU-Sensor/die IMU-Sensoren 1766 das Fahrzeug 1700 in die Lage versetzen, den Kurs zu schätzen, ohne dass Eingaben von einem Magnetsensor erforderlich sind, indem die Geschwindigkeitsänderungen von GPS direkt beobachtet und mit dem/den IMU-Sensor(en) 1766 korreliert werden. In einigen Beispielen können der/die IMU-Sensor (en) 1766 und der/die GNSS-Sensor(en) 1758 in einer einzigen integrierten Einheit kombiniert werden.

[0213] Das Fahrzeug kann Mikrofon(e) 1796 enthalten, die im und/oder um das Fahrzeug 1700 herum angebracht sind. Das/die Mikrofon(e) 1796 können u. a. zur Erkennung und Identifizierung von Einsatzfahrzeugen verwendet werden.

[0214] Das Fahrzeug kann ferner eine beliebige Anzahl von Kameratypen enthalten, einschließlich Stereokamera(s) 1768, Weitwinkelkamera(s) 1770, Infrarotkamera(s) 1772, Surround-Kamera(s) 1774, Fern- und/oder Mittelbereichskamera(s) 1798 und/oder andere Kameratypen. Die Kameras können verwendet werden, um Bilddaten rund um den gesamten Umfang des Fahrzeugs 1700 zu erfassen. Die verwendeten Kameratypen hängen von den Ausführungsformen und Anforderungen für das Fahrzeug 1700 ab, und es kann eine beliebige Kombination von Kameratypen verwendet werden, um die erforderliche Abdeckung um das Fahrzeug 1700 herum zu gewährleisten. Auch die Anzahl der Kameras kann je nach Ausführungsform unterschiedlich sein. Beispielsweise kann das Fahrzeug sechs Kameras, sieben Kameras, zehn Kameras, zwölf Kameras und/oder eine andere Anzahl von Kameras umfassen. Die Kameras können zum Beispiel und ohne Einschränkung Gigabit Multimedia Serial Link (GMSL) und/oder Gigabit Ethernet unterstützen.

Jede der Kameras wird hierin in Bezug auf **Fig. 17A** und **Fig. 17B** ausführlicher beschrieben.

[0215] Das Fahrzeug 1700 kann außerdem einen oder mehrere Schwingungssensoren 1742 enthalten. Der/die Schwingungssensor(en) 1742 können Schwingungen von Komponenten des Fahrzeugs, wie z. B. der Achse(n), messen. So können beispielsweise Änderungen der Schwingungen auf eine Änderung der Straßenoberfläche hinweisen. In einem anderen Beispiel können bei Verwendung von zwei oder mehr Schwingungssensoren 1742 die Unterschiede zwischen den Schwingungen verwendet werden, um die Reibung oder den Schlupf der Straßenoberfläche zu bestimmen (z. B. wenn der Unterschied in der Schwingung zwischen einer angetriebenen Achse und einer frei drehenden Achse besteht).

[0216] Das Fahrzeug 1700 kann ein ADAS-System 1738 enthalten. Das ADAS-System 1738 kann in einigen Beispielen einen SoC enthalten. Das ADAS-System 1738 kann einen autonomen/adaptiven/automatischen Geschwindigkeitsregler (ACC), einen kooperativen adaptiven Geschwindigkeitsregler (CACC), eine Auffahrwarnung (FCW), eine automatische Notbremsung (AEB), eine Spurverlassenswarnung (LDW), einen Spurhalteassistenten (LKA), einen Toter-Winkel-Warner (BSW), einen Querverkehrswarner (RCTW), ein Kollisionswarnsystem (CWS), eine Fahrspurzentrierung (LC) und/oder andere Merkmale und Funktionen umfassen.

[0217] Die ACC-Systeme können RADAR-Sensor(en) 1760, LIDAR-Sensor(en) 1764 und/oder eine Kamera(en) verwenden. Die ACC-Systeme können einen ACC in Längsrichtung und/oder einen ACC in Querrichtung umfassen. Der ACC in Längsrichtung überwacht und steuert den Abstand zu dem unmittelbar vor dem Fahrzeug 1700 befindlichen Fahrzeug und passt die Fahrzeuggeschwindigkeit automatisch an, um einen sicheren Abstand zu vorausfahrenden Fahrzeugen einzuhalten. Der seitliche ACC hält den Abstand und weist das Fahrzeug 1700 an, die Spur zu wechseln, wenn dies erforderlich ist. Lateral ACC ist mit anderen ADAS-Anwendungen wie LCA und CWS verwandt.

[0218] CACC verwendet Informationen von anderen Fahrzeugen, die über die Netzwerkschnittstelle 1724 und/oder die Funkantenne(n) 1726 von anderen Fahrzeugen über eine drahtlose Verbindung oder indirekt über eine Netzwerkverbindung (z. B. über das Internet) empfangen werden können. Direkte Verbindungen können über eine Fahrzeug-zu-Fahrzeug-Kommunikationsverbindung (V2V) hergestellt werden, während indirekte Verbindungen eine Infrastruktur-zu-Fahrzeug-Kommunikationsverbindung (12V) sein können. Im Allgemeinen liefert das V2V-Kommunikationskonzept Informationen über die unmittelbar vorausfahrenden Fahrzeuge (z. B. Fahrzeuge, die sich unmittelbar vor dem Fahrzeug 1700 und auf derselben Fahrspur wie dieses befinden), während das I2 V-Kommunikationskonzept Informationen über den weiter vorausfahrenden Verkehr liefert. CACC-Systeme können sowohl I2V- als auch V2V-Informationsquellen enthalten. Angesichts der Informationen über die vor dem Fahrzeug 1700 fahrenden Fahrzeuge kann das CACC-System zuverlässiger sein und hat das Potenzial, den Verkehrsfluss zu verbessern und Staus auf der Straße zu verringern.

[0219] FCW-Systeme sind so konzipiert, dass sie den Fahrer vor einer Gefahr warnen, so dass er korrigierend eingreifen kann. FCW-Systeme verwenden eine nach vorne gerichtete Kamera und/oder RADAR-Sensor(en) 1760, die mit einem speziellen Prozessor, DSP, FPGA und/oder ASIC gekoppelt sind, der elektrisch mit der Rückmeldung an den Fahrer verbunden ist, z. B. mit einem Display, einem Lautsprecher und/oder einer vibrierenden Komponente. FCW-Systeme können eine Warnung ausgeben, z. B. in Form eines Tons, einer optischen Warnung, einer Vibration und/oder eines schnellen Bremsimpulses.

[0220] AEB-Systeme erkennen einen drohenden Zusammenstoß mit einem anderen Fahrzeug oder einem anderen Objekt und können automatisch die Bremsen betätigen, wenn der Fahrer nicht innerhalb eines bestimmten Zeit- oder Entfernungsparameters korrigierend eingreift. AEB-Systeme können nach vorne gerichtete Kamera(s) und/oder RADAR-Sensoren 1760 verwenden, die mit einem speziellen Prozessor, DSP, FPGA und/oder ASIC gekoppelt sind. Wenn das AEB-System eine Gefahr erkennt, warnt es in der Regel zunächst den Fahrer, damit er korrigierend eingreift, um die Kollision zu vermeiden. Wenn der Fahrer keine korrigierenden Maßnahmen ergreift, kann das AEB-System automatisch die Bremsen betätigen, um die Auswirkungen der vorhergesagten Kollision zu verhindern oder mindestens abzumildern. AEB-Systeme können Techniken wie die dynamische Bremsunterstützung und/oder die Crash-Imminent Braking umfassen.

[0221] Spurhalteassistenten warnen den Fahrer durch optische, akustische und/oder taktile Signale, z. B. durch Vibrationen am Lenkrad oder am Sitz, wenn das Fahrzeug 1700 die Fahrbahnmarkierungen überfährt. Ein Spurhaltewarnsystem wird nicht aktiviert, wenn der Fahrer durch Betätigen des Blinkers ein absichtliches Verlassen der Fahrspur anzeigt. LDW-Systeme können nach vorne gerichtete Kameras verwenden, die mit einem speziellen Prozessor, DSP, FPGA und/oder ASIC verbunden sind, der elektrisch mit dem Fahrerfeedback gekoppelt ist, z. B. mit einer Anzeige, einem Lautsprecher und/oder einer vibrierenden Komponente.

[0222] LKA-Systeme sind eine Variante der Spurhaltewarnsysteme. LKA-Systeme sorgen für einen Lenkeingriff oder eine Bremsung, um das Fahrzeug 1700 zu korrigieren, wenn das Fahrzeug 1700 beginnt, die Fahrspur zu verlassen.

[0223] BSW-Systeme erkennen und warnen den Fahrer vor Fahrzeugen im toten Winkel des Fahrzeugs. BSW-Systeme können ein optisches, akustisches und/oder taktiles Warnsignal ausgeben, um darauf hinzuweisen, dass das Zusammenführen oder Wechseln der Fahrspur unsicher ist. Das System kann eine zusätzliche Warnung ausgeben, wenn der Fahrer einen Blinker betätigt. BSW-Systeme können nach hinten gerichtete Kamera(s) und/oder RADAR-Sensor(en) 1760 verwenden, die mit einem speziellen Prozessor, DSP, FPGA und/oder ASIC gekoppelt sind, der elektrisch mit dem Fahrerfeedback gekoppelt ist, z. B. mit einem Display, Lautsprecher und/oder einer vibrierenden Komponente.

[0224] RCTW-Systeme können eine visuelle, akustische und/oder taktile Benachrichtigung ausgeben, wenn beim Rückwärtsfahren des Fahrzeugs 1700 ein Objekt außerhalb der Reichweite der Rückfahrkamera erkannt wird. Einige RCTW-Systeme umfassen AEB, um sicherzustellen, dass die Fahrzeugbremsen betätigt werden, um einen Unfall zu vermeiden. RCTW-Systeme können einen oder mehrere nach hinten gerichtete RADAR-Sensoren 1760 verwenden, die mit einem speziellen Prozessor, DSP, FPGA und/oder ASIC gekoppelt sind, der elektrisch mit dem Fahrerfeedback gekoppelt ist, z. B. mit einem Display, Lautsprecher und/oder einer vibrierenden Komponente.

[0225] Bei herkömmlichen ADAS-Systemen kann es zu falsch-positiven Ergebnissen kommen, die für den Fahrer ärgerlich und ablenkend sein können, aber in der Regel nicht katastrophal sind, weil die ADAS-Systeme den Fahrer warnen und es ihm ermöglichen, zu entscheiden, ob wirklich eine Sicherheitsbedingung vorliegt und entsprechend zu handeln. In einem autonomen Fahrzeug 1700 muss das Fahrzeug 1700 jedoch im Falle widersprüchlicher Ergebnisse selbst entscheiden, ob es das Ergebnis eines primären Computers oder eines sekundären Computers (z. B. eines ersten Controllers 1736 oder eines zweiten Controllers 1736) beherzigen soll. In einigen Ausführungsformen kann das ADAS-System 1738 beispielsweise ein Backup- und/oder Sekundärcomputer sein, der Wahrnehmungsinformationen an ein Rationalitätsmodul des Backup-Computers liefert. Der Rationalitätsmonitor des Backup-Rechners kann eine redundante, diverse Software auf Hardwarekomponenten ausführen, um Fehler in der Wahrnehmung und bei dynamischen Fahraufgaben zu erkennen. Die Ausgaben des ADAS-Systems 1738 können an eine übergeordnete MCU weitergeleitet werden. Wenn die Ausgaben des Primärrechners und des Sekundärrechners kollidieren, muss die überwachende MCU bestimmen, wie der Konflikt gelöst werden kann, um einen sicheren Betrieb zu gewährleisten.

[0226] In einigen Beispielen kann der primäre Computer so konfiguriert sein, dass er der übergeordneten MCU einen Konfidenzwert liefert, der das Vertrauen des primären Computers in das gewählte Ergebnis angibt. Übersteigt der Konfidenzwert einen Schwellenwert, kann die überwachende MCU der Anweisung des Primärcomputers folgen, unabhängig davon, ob der Sekundärcomputer ein widersprüchliches oder inkonsistentes Ergebnis liefert. Erreicht der Konfidenzwert den Schwellenwert nicht und zeigen der primäre und der sekundäre Computer unterschiedliche Ergebnisse an (z. B. den Konflikt), kann die übergeordnete MCU zwischen den Computern vermitteln, um das geeignete Ergebnis zu bestimmen.

[0227] Die überwachende MCU kann so konfiguriert sein, dass sie ein neuronales Netz bzw. neuronale Netze ausführt, das bzw. die so trainiert und konfiguriert ist bzw. sind, dass es bzw. sie basierend auf der Ausgaben des Primärcomputers und des Sekundärcomputers die Bedingungen bestimmt bzw. bestimmen, unter denen der Sekundärcomputer Fehlalarme auslöst. So kann das neuronale Netz bzw. können die neuronalen Netze in der Überwachungs-MCU lernen, wann der Ausgabe des Sekundärcomputers vertraut werden kann und wann nicht. Handelt es sich bei dem sekundären Computer beispielsweise um ein RADAR-basiertes FCW-System, kann ein neuronales Netz in der übergeordneten MCU lernen, wann das FCW-System metallische Objekte identifiziert, die in Wirklichkeit keine Gefahr darstellen, wie etwa ein Abflussgitter oder ein Schachtdeckel, der einen Alarm auslöst. Wenn der Sekundärcomputer ein kamerabasiertes Spurhaltewarnsystem ist, kann ein neuronales Netz in der Überwachungs-MCU lernen, das Spurhaltewarnsystem außer Kraft zu setzen, wenn Radfahrer oder Fußgänger vorhanden sind und ein Verlassen der Fahrspur tatsächlich das sicherste Manöver ist. In Ausführungsformen, in denen ein neuronales Netz bzw. neuronale Netze auf der überwachenden MCU laufen, kann die überwachende MCU mindestens einen DLA oder eine GPU enthalten, der für den Betrieb des neuronalen Netzes bzw. der neuronalen Netze mit zugehörigem Speicher geeignet ist. In bevorzugten Ausführungsformen kann die Überwachungs-MCU eine Komponente des/der SoC(s) 1704 umfassen und/oder als solche enthalten sein.

[0228] In anderen Beispielen kann das ADAS-System 1738 einen sekundären Computer enthalten, der die ADAS-Funktionen unter Verwendung herkömmlicher Regeln der Computer Vision ausführt. Der sekundäre Computer kann also klassische Regeln

des Computersehens verwenden (wenn-dann), und das Vorhandensein eines neuronalen Netzes bzw. neuronaler Netze in der übergeordneten MCU kann die Zuverlässigkeit, Sicherheit und Leistung verbessern. Beispielsweise wird das Gesamtsystem durch die unterschiedliche Implementierung und die absichtliche Nichtidentität fehlertoleranter, insbesondere gegenüber Fehlern, die durch Softwarefunktionen (oder Software-Hardware-Schnittstellen) verursacht werden. Wenn beispielsweise ein Softwarefehler in der auf dem Primärcomputer laufenden Software auftritt und der nicht identische Softwarecode auf dem Sekundärcomputer das gleiche Gesamtergebnis liefert, kann die überwachende MCU mit größerer Sicherheit davon ausgehen, dass das Gesamtergebnis korrekt ist und der Fehler in der Software oder Hardware des Primärcomputers keinen wesentlichen Fehler verursacht.

[0229] In einigen Beispielen kann die Ausgabe des ADAS-Systems 1738 in den Wahrnehmungsblock des Primärrechners und/oder in den Block für dynamische Fahraufgaben des Primärrechners eingespeist werden. Wenn das ADAS-System 1738 beispielsweise eine Warnung vor einem Aufprall aufgrund eines unmittelbar vorausliegenden Objekts anzeigt, kann der Wahrnehmungsblock diese Information bei der Identifizierung von Objekten verwenden. In anderen Beispielen kann der Sekundärcomputer über ein eigenes neuronales Netz verfügen, das trainiert ist und somit das Risiko von Fehlalarmen verringert, wie hier beschrieben.

[0230] Das Fahrzeug 1700 kann ferner das Infotainment-SoC 1730 (z. B. ein bordeigenes Infotainment-System (IVI)) enthalten. Obwohl als SoC dargestellt und beschrieben, kann das Infotainment-System kein SoC sein und zwei oder mehr diskrete Komponenten umfassen. Das Infotainment-SoC 1730 kann eine Kombination aus Hardware und Software enthalten, die zur Bereitstellung von Audio (z. B. Musik, einem persönlichen digitalen Assistenten, Navigationsanweisungen, Nachrichten, Radio usw.), Video (z. B. Fernsehen, Filme, Streaming usw.), Telefon (z. B. Freisprechen), Netzwerkkonnektivität (z. B., LTE, Wi-Fi usw.) und/oder Informationsdienste (z. B. Navigationssysteme, Einparkhilfe hinten, ein Radiodatensystem, fahrzeugbezogene Informationen wie Kraftstoffstand, zurückgelegte Gesamtstrecke, Bremskraftstoffstand, Ölstand, Tür öffnen/-schließen, Luftfilterinformationen usw.) an das Fahrzeug 1700. Der Infotainment-SoC 1730 kann beispielsweise Radios, Plattenspieler, Navigationssysteme, Videoplayer, USB- und Bluetooth-Konnektivität, Carputer, In-Car-Entertainment, Wi-Fi, Audiobedienelemente am Lenkrad, Freisprecheinrichtung, ein Heads-up-Display (HUD), ein HMI-Display 1734, ein Telematikgerät, ein Bedienfeld (z. B. zur Steuerung und/oder Interaktion mit verschiedenen Komponenten, Funktionen und/oder Systemen) und/oder andere Komponenten umfassen. Der Infotainment-SoC 1730 kann ferner verwendet werden, um einem oder mehreren Nutzern des Fahrzeugs Informationen (z. B. visuell und/oder akustisch) bereitzustellen, wie etwa Informationen vom ADAS-System 1738, Informationen zum autonomen Fahren wie geplante Fahrzeugmanöver, Trajektorien, Umgebungsinformationen (z. B. Kreuzungsinformationen, Fahrzeuginformationen, Straßeninformationen usw.) und/oder andere Informationen.

[0231] Der Infotainment-SoC 1730 kann GPU-Funktionen enthalten. Der Infotainment-SoC 1730 kann über den Bus 1702 (z. B. CAN-Bus, Ethernet usw.) mit anderen Geräten, Systemen und/oder Komponenten des Fahrzeugs 1700 kommunizieren. In einigen Beispielen kann der Infotainment-SoC 1730 mit einer Überwachungs-MCU gekoppelt sein, so dass die GPU des Infotainment-Systems einige Selbstfahrfunktionen ausführen kann, falls die primäre(n) Steuereinheit(en) 1736 (z. B. die primären und/oder Backup-Computer des Fahrzeugs 1700) ausfallen. In einem solchen Beispiel kann der Infotainment-SoC 1730 das Fahrzeug 1700 in einen Chauffeurmodus bis zum sicheren Anhalten versetzen, wie hier beschrieben.

[0232] Das Fahrzeug 1700 kann ferner ein Kombiinstrument 1732 (z. B. ein digitales Armaturenbrett, ein elektronisches Kombiinstrument, eine digitale Instrumententafel usw.) enthalten. Das Kombiinstrument 1732 kann ein Controller und/oder einen Supercomputer (z. B. ein diskretes Controller oder einen Supercomputer) enthalten. Das Kombiinstrument 1732 kann eine Reihe von Instrumenten enthalten, wie z. B. Tachometer, Kraftstoffstand, Öldruck, Drehzahlmesser, Kilometerzähler, Blinker, Schaltstellungsanzeige, Sicherheitsgurt-Warnleuchte(n), Parkbrems-Warnleuchte(n), Motor-Fehlfunktionsleuchte (n), Informationen über das Airbag-System (SRS), Beleuchtungssteuerungen, Sicherheitssystemsteuerungen, Navigationsinformationen usw. In einigen Beispielen können Informationen vom Infotainment SoC 1730 und dem Kombiinstrument 1732 angezeigt und/oder gemeinsam genutzt werden. Mit anderen Worten: Das Kombiinstrument 1732 kann Teil des Infotainment-SoC 1730 sein oder umgekehrt.

[0233] Fig. 17D ist ein Systemdiagramm für die Kommunikation zwischen dem/den Cloud-basierten Server(n) und dem autonomen Beispielfahrzeug 1700 aus **Fig. 17A**, gemäß einigen Ausführungsformen der vorliegenden Offenbarung. Das System 1776 kann Server 1778, Netzwerk(e) 1790 und Fahrzeuge, einschließlich des Fahrzeugs 1700, umfassen. Der (die) Server 1778 können eine Vielzahl von GPUs 1784(A)-1784(H) (hier zusammenfassend als GPUs 1784 bezeichnet), PCIe-Switches 1782 (A)-1782(H) (hier zusammenfassend als PCIe-Switches 1782 bezeichnet) und/oder CPUs 1780

(A)-1780(B) (hier zusammenfassend als CPUs 1780 bezeichnet) umfassen. Die GPUs 1784, die CPUs 1780 und die PCIe-Switches können über Hochgeschwindigkeitsverbindungen miteinander verbunden sein, wie z. B. und ohne Einschränkung die von NVIDIA entwickelten NVLink-Schnittstellen 1788 und/oder PCIe-Verbindungen 1786. In einigen Beispielen sind die GPUs 1784 über NVLink und/oder NVSwitch SoC und die GPUs 1784 und die PCIe-Switches 1782 über PCIe-Verbindungen verbunden. Obwohl acht GPUs 1784, zwei CPUs 1780 und zwei PCIe-Switches abgebildet sind, ist dies nicht als Einschränkung zu verstehen. Je nach Ausführungsform kann jeder der Server 1778 eine beliebige Anzahl von GPUs 1784, CPUs 1780 und/oder PCIe-Switches umfassen. Beispielsweise können die Server 1778 jeweils acht, sechzehn, zweiunddreißig und/oder mehr GPUs 1784 enthalten.

[0234] Der (die) Server 1778 können über das (die) Netzwerk(e) 1790 und von den Fahrzeugen Bilddaten empfangen, die für Bilder repräsentativ sind, die unerwartete oder veränderte Straßenzustände zeigen, z. B. kürzlich begonnene Straßenarbeiten. Der (die) Server 1778 können über das (die) Netzwerk(e) 1790 und an die Fahrzeuge neuronale Netze 1792, aktualisierte neuronale Netze 1792 und/oder Karteninformationen 1794, einschließlich Informationen über Verkehrs- und Straßenbedingungen, übertragen. Die Aktualisierungen der Karteninformationen 1794 können Aktualisierungen für die HD-Karte 1722 enthalten, wie beispielsweise Informationen über Baustellen, Schlaglöcher, Umleitungen, Überschwemmungen und/oder andere Hindernisse. In einigen Beispielen können die neuronalen Netze 1792, die aktualisierten neuronalen Netze 1792 und/oder die Karteninformationen 1794 aus neuem Training und/oder Erfahrungen resultieren, die in Daten dargestellt sind, die von einer beliebigen Anzahl von Fahrzeugen in der Umgebung empfangen wurden, und/oder auf einem Training basieren, das in einem Datenzentrum durchgeführt wurde (z. B. unter Verwendung des/der Server(s) 1778 und/oder anderer Server).

[0235] Der/die Server 1778 können verwendet werden, um Modelle für maschinelles Lernen (z. B. neuronale Netze) basierend auf Trainingsdaten zu trainieren. Die Trainingsdaten können von den Fahrzeugen und/oder in einer Simulation (z. B. mit einer Spiel-Engine) erzeugt werden. In einigen Beispielen werden die Trainingsdaten mit Tags versehen (z. B. wenn das neuronale Netz von überwachtem Lernen profitiert) und/oder anderen Vorverarbeitungen unterzogen, während in anderen Beispielen die Trainingsdaten nicht mit Tags versehen und/oder vorverarbeitet werden (z. B. wenn das neuronale Netz kein überwachtes Lernen benötigt). Das Training kann nach einer oder mehreren Klassen von maschinellen Lerntechniken durchgeführt werden, einschließlich, aber nicht beschränkt auf Klassen wie: überwachtes Training, halbüberwachtes Training, unüberwachtes Training, selbstlernendes Lernen, Verstärkungslernen, föderiertes Lernen, Transferlernen, Merkmalslemen (einschließlich Hauptkomponenten- und Clusteranalysen), multilineares Unterraumlemen, vielfältiges Lernen, Repräsentationslemen (einschließlich Lernen mit Ersatzwörterbüchern), regelbasiertes maschinelles Lernen, Anomalieerkennung und alle Varianten oder Kombinationen davon. Sobald die maschinellen Lernmodelle trainiert sind, können die maschinellen Lernmodelle von den Fahrzeugen verwendet werden (z. B. durch Übertragung an die Fahrzeuge über das/die Netzwerk(e) 1790 und/oder die maschinellen Lernmodelle können von dem/den Server(n) 1778 zur Fernüberwachung der Fahrzeuge verwendet werden.

[0236] In einigen Beispielen können die Server 1778 Daten von den Fahrzeugen empfangen und die Daten auf aktuelle neuronale Netze in Echtzeit für intelligente Inferenzen in Echtzeit anwenden. Der/die Server 1778 können Deep-Learning-Supercomputer und/oder dedizierte KI-Computer umfassen, die von GPU(s) 1784 angetrieben werden, wie z. B. die von NVIDIA entwickelten DGX- und DGX-Station-Maschinen. In einigen Beispielen können die Server 1778 jedoch auch Deep-Learning-Infrastrukturen umfassen, die nur CPU-betriebene Rechenzentren verwenden.

[0237] Die Deep-Learning-Infrastruktur des/der Server(s) 1778 kann zur schnellen Inferenz in Echtzeit fähig sein und kann diese Fähigkeit nutzen, um den Zustand der Prozessoren, der Software und/oder der zugehörigen Hardware im Fahrzeug 1700 zu bewerten und zu überprüfen. Zum Beispiel kann die Deep-Learning-Infrastruktur periodische Updates vom Fahrzeug 1700 erhalten, wie z. B. eine Sequenz von Bildern und/oder Objekten, die das Fahrzeug 1700 in dieser Sequenz von Bildern lokalisiert hat (z. B. über Computer Vision und/oder andere maschinelle Objektklassifizierungstechniken). Die Deep-Learning-Infrastruktur kann ihr eigenes neuronales Netz laufen lassen, um die Objekte zu identifizieren und sie mit den vom Fahrzeug 1700 identifizierten Objekten zu vergleichen. Wenn die Ergebnisse nicht übereinstimmen und die Infrastruktur zu dem Schluss kommt, dass die KI im Fahrzeug 1700 nicht richtig funktioniert, kann der Server 1778 ein Signal an das Fahrzeug 1700 senden, das einen ausfallsicheren Computer des Fahrzeugs 1700 anweist, die Kontrolle zu übernehmen, die Fahrgäste zu benachrichtigen und ein sicheres Parkmanöver durchzuführen.

[0238] Für die Inferenzerstellung kann der/die Server 1778 die GPU(s) 1784 und einen oder mehrere programmierbare Inferenzbeschleuniger (z. B. Ten-

sorRT von NVIDIA) enthalten. Die Kombination von GPU-gesteuerten Servern und Inferenzbeschleunigung kann eine Reaktionsfähigkeit in Echtzeit ermöglichen. In anderen Beispielen, in denen die Leistung weniger kritisch ist, können Server mit CPUs, FPGAs und anderen Prozessoren für die Inferenzbildung verwendet werden.

BEISPIEL FÜR EINE DATENVERARBEITUNGSANLAGE

[0239] **Fig. 18** ist ein Blockdiagramm einer beispielhaften Rechenvorrichtung(en) 1800, die zur Verwendung bei der Implementierung einiger Ausführungsformen der vorliegenden Offenbarung geeignet ist. Die Rechenvorrichtung 1800 kann ein Verbindungssystem 1802 umfassen, das die folgenden Vorrichtungen direkt oder indirekt koppelt: Speicher 1804, eine oder mehrere Zentraleinheiten (CPUs) 1806, eine oder mehrere Grafikverarbeitungseinheiten (GPUs) 1808, eine Kommunikationsschnittstelle 1810, Ein-/Ausgabeanschlüsse (I/O) 1812, Ein-/Ausgabekomponenten 1814, eine Stromversorgung 1816, eine oder mehrere Präsentationskomponenten 1818 (z. B. Anzeige(n)) und eine oder mehrere Logikeinheiten 1820. In mindestens einer Ausführungsform kann die Rechenvorrichtung(en) 1800 eine oder mehrere virtuelle Maschinen (VMs) umfassen und/oder jede ihrer Komponenten kann virtuelle Komponenten (z. B. virtuelle Hardwarekomponenten) umfassen. Als nicht einschränkende Beispiele können eine oder mehrere der GPUs 1808 eine oder mehrere vGPUs umfassen, eine oder mehrere der CPUs 1806 können eine oder mehrere vCPUs umfassen und/oder eine oder mehrere der Logikeinheiten 1820 können eine oder mehrere virtuelle Logikeinheiten umfassen. Von daher können ein (mehrere) Rechengerät(e) 1800 diskrete Komponenten (z. B. eine vollständige GPU, die dem Rechengerät 1800 gewidmet ist), virtuelle Komponenten (z. B. einen Teil einer GPU, die dem Rechengerät 1800 gewidmet ist) oder eine Kombination davon umfassen.

[0240] Obwohl die verschiedenen Blöcke in **Fig. 18** als über das Verbindungssystem 1802 mit Leitungen verbunden dargestellt sind, ist dies nicht als Einschränkung zu verstehen und dient nur der Klarheit. In einigen Ausführungsformen kann zum Beispiel eine Präsentationskomponente 1818, wie ein Anzeigegerät, als I/O-Komponente 1814 betrachtet werden (z. B. wenn die Anzeige ein Touchscreen ist). Als weiteres Beispiel können die CPUs 1806 und/oder GPUs 1808 Speicher enthalten (z. B. kann der Speicher 1804 zusätzlich zum Speicher der GPUs 1808, der CPUs 1806 und/oder anderer Komponenten ein Speichergerät darstellen). Mit anderen Worten, die Rechenvorrichtung der **Fig. 18** ist lediglich illustrativ. Es wird nicht zwischen Kategorien wie „Workstation“, „Server“, „Laptop“, „Desktop“, „Tablet“, „Client-Gerät“, „mobiles Gerät“, „Handheld-Gerät“, „Spielkonsole“, „elektronische Steuereinheit (ECU)“, „Virtual-Reality-System“ und/oder anderen Geräte- oder Systemtypen unterschieden, da sie alle in den Anwendungsbereich der Rechenvorrichtung der **Fig. 18** fallen.

[0241] Das Verbindungssystem 1802 kann eine oder mehrere Verbindungen oder Busse darstellen, wie z. B. einen Adressbus, einen Datenbus, einen Steuerbus oder eine Kombination davon. Das Verbindungssystem 1802 kann einen oder mehrere Bus- oder Verbindungstypen umfassen, z. B. einen ISA-Bus (Industry Standard Architecture), einen EISA-Bus (Extended Industry Standard Architecture), einen VESA-Bus (Video Electronics Standards Association), einen PCI-Bus (Peripheral Component Interconnect), einen PCIe-Bus (Peripheral Component Interconnect Express) und/oder eine andere Art von Bus oder Verbindung. In einigen Ausführungsformen gibt es direkte Verbindungen zwischen den Komponenten. Zum Beispiel kann die CPU 1806 direkt mit dem Speicher 1804 verbunden sein. Außerdem kann die CPU 1806 direkt mit der GPU 1808 verbunden sein. Bei direkten oder Punkt-zu-Punkt-Verbindungen zwischen Komponenten kann das Verbindungssystem 1802 eine PCIe-Verbindung enthalten, um die Verbindung herzustellen. In diesen Beispielen muss ein PCI-Bus nicht in der Rechenvorrichtung 1800 enthalten sein.

[0242] Der Speicher 1804 kann aus einer Vielzahl von computerlesbaren Medien bestehen. Bei den computerlesbaren Medien kann es sich um alle verfügbaren Medien handeln, auf die das Computergerät 1800 zugreifen kann. Die computerlesbaren Medien können sowohl flüchtige als auch nicht-flüchtige Medien sowie entfernbare und nicht-entfernbare Medien umfassen. Als Beispiel und ohne Einschränkung können die computerlesbaren Medien Computerspeichermedien und Kommunikationsmedien umfassen.

[0243] Die Computerspeichermedien können sowohl flüchtige als auch nichtflüchtige Medien und/oder entfernbare und nicht entfernbare Medien umfassen, die in einem beliebigen Verfahren oder einer beliebigen Technologie zur Speicherung von Informationen wie computerlesbaren Anweisungen, Datenstrukturen, Programmmodulen und/oder anderen Datentypen implementiert sind. Zum Beispiel kann der Speicher 1804 computerlesbare Anweisungen speichern (z. B., die ein oder mehrere Programme und/oder ein oder mehrere Programmelemente darstellen, wie z. B. ein Betriebssystem. Zu den Computerspeichermedien können unter anderem RAM, ROM, EEPROM, Flash-Speicher oder andere Speichertechnologien, CD-ROM, Digital Versatile Disks (DVD) oder andere optische Plattenspeicher, Magnetkassetten, Magnetbänder, Magnetplat-

tenspeicher oder andere magnetische Speichervorrichtungen oder jedes andere Medium gehören, das zur Speicherung der gewünschten Informationen verwendet werden kann und auf das die Rechenvorrichtung 1800 zugreifen kann. Wie hierin verwendet, umfassen Computerspeichermedien nicht per se Signale.

[0244] Die Computerspeichermedien können computerlesbare Befehle, Datenstrukturen, Programmmodule und/oder andere Datentypen in einem modulierten Datensignal wie z. B. einer Trägerwelle oder einem anderen Transportmechanismus verkörpern und umfassen beliebige Informationsübertragungsmedien. Der Begriff „moduliertes Datensignal“ kann sich auf ein Signal beziehen, bei dem eine oder mehrere seiner Eigenschaften so eingestellt oder verändert sind, dass Informationen in dem Signal kodiert werden. Die Computerspeichermedien können beispielsweise verdrahtete Medien, wie ein verdrahtetes Netzwerk oder eine direkte Kabelverbindung, und drahtlose Medien, wie akustische, RF-, Infrarot- und andere drahtlose Medien, umfassen. Kombinationen der oben genannten Medien sollten ebenfalls in den Bereich der computerlesbaren Medien fallen.

[0245] Die CPU(s) 1806 können so konfiguriert sein, dass sie mindestens einige der computerlesbaren Anweisungen ausführen, um eine oder mehrere Komponenten der Rechenvorrichtung 1800 zu steuern, um eines oder mehrere der hierin beschriebenen Verfahren und/oder Prozesse durchzuführen. Die CPU(s) 1806 können jeweils einen oder mehrere Kerne (z. B. einen, zwei, vier, acht, achtundzwanzig, zweiundsiebzig usw.) umfassen, die in der Lage sind, eine Vielzahl von Software-Threads gleichzeitig zu bearbeiten. Die CPU(s) 1806 können jede Art von Prozessor umfassen und je nach Art des implementierten Rechenvorrichtung 1800 verschiedene Arten von Prozessoren umfassen (z. B. Prozessoren mit weniger Kernen für mobile Geräte und Prozessoren mit mehr Kernen für Server). Je nach Art der Rechenvorrichtung 1800 kann der Prozessor beispielsweise ein Advanced RISC Machines (ARM)-Prozessor sein, der mit Reduced Instruction Set Computing (RISC) arbeitet, oder ein x86-Prozessor, der mit Complex Instruction Set Computing (CISC) arbeitet. Die Rechenvorrichtung 1800 kann eine oder mehrere CPUs 1806 zusätzlich zu einem oder mehreren Mikroprozessoren oder zusätzlichen Coprozessoren, wie z. B. mathematischen Coprozessoren, enthalten.

[0246] Zusätzlich zu oder alternativ zu der/den CPU(s) 1806 können die GPU(s) 1808 so konfiguriert sein, dass sie mindestens einige der computerlesbaren Anweisungen ausführen, um eine oder mehrere Komponenten der Rechenvorrichtung 1800 zu steuern, um eines oder mehrere der hier beschriebenen Verfahren und/oder Prozesse durchzuführen. Eine oder mehrere der GPU(s) 1808 können eine integrierte GPU sein (z. B. mit einer oder mehreren der CPU(s) 1806 und/oder eine oder mehrere der GPU(s) 1808 können eine diskrete GPU sein. In Ausführungsformen kann eine oder mehrere der GPU(s) 1808 ein Coprozessor einer oder mehrerer der CPU(s) 1806 sein. Der/die GPU(s) 1808 können von der Rechenvorrichtung 1800 verwendet werden, um Grafiken (z. B. 3D-Grafiken) zu rendern oder allgemeine Berechnungen durchzuführen. Die GPU(s) 1808 können zum Beispiel für General-Purpose Computing on GPUs (GPGPU) verwendet werden. Die GPU(s) 1808 können Hunderte oder Tausende von Kernen umfassen, die in der Lage sind, Hunderte oder Tausende von Software-Threads gleichzeitig zu verarbeiten. Die GPU(s) 1808 können als Reaktion auf Rendering-Befehle (z. B. Rendering-Befehle von der/den CPU(s) 1806, die über eine Host-Schnittstelle empfangen werden) Pixeldaten für Ausgabebilder erzeugen. Die GPU(s) 1808 können einen Grafikspeicher, wie z. B. einen Anzeigespeicher, zum Speichern von Pixeldaten oder anderen geeigneten Daten, wie z. B. GPGPU-Daten, enthalten. Der Anzeigespeicher kann als Teil des Speichers 1804 enthalten sein. Die GPU(s) 1808 können zwei oder mehr GPUs umfassen, die parallel arbeiten (z. B. über eine Verbindung). Die Verbindung kann die GPUs direkt verbinden (z. B. mit NVLINK) oder über einen Switch (z. B. mit NVSwitch). Wenn sie miteinander kombiniert werden, kann jede GPU 1808 Pixeldaten oder GPGPU-Daten für verschiedene Teile einer Ausgabe oder für verschiedene Ausgaben erzeugen (z. B. eine erste GPU für ein erstes Bild und eine zweite GPU für ein zweites Bild). Jede GPU kann ihren eigenen Speicher haben oder sich den Speicher mit anderen GPUs teilen.

[0247] Zusätzlich zu oder alternativ zu der (den) CPU(s) 1806 und/oder der (den) GPU(s) 1808 können die Logikeinheit(en) 1820 so konfiguriert sein, dass sie mindestens einige der computerlesbaren Anweisungen ausführt (ausführen), um eine oder mehrere Komponenten der Rechenvorrichtung 1800 zu steuern, um eines oder mehrere der hier beschriebenen Verfahren und/oder Prozesse durchzuführen. In Ausführungsformen können die CPU(s) 1806, die GPU(s) 1808 und/oder die Logikeinheit(en) 1820 diskret oder gemeinsam eine beliebige Kombination der Verfahren, Prozesse und/oder Teile davon ausführen. Eine oder mehrere der Logikeinheiten 1820 können Teil einer oder mehrerer der CPU(s) 1806 und/oder der GPU(s) 1808 sein und/oder eine oder mehrere der Logikeinheiten 1820 können diskrete Komponenten oder anderweitig außerhalb der CPU(s) 1806 und/oder der GPU(s) 1808 sein. In Ausführungsformen kann eine oder mehrere der Logikeinheiten 1820 ein Coprozessor einer oder mehrerer der CPU(s) 1806 und/oder einer oder mehrerer der GPU(s) 1808 sein.

[0248] Beispiele für die Logikeinheit(en) 1820 umfassen einen oder mehrere Verarbeitungskerne und/oder Komponenten davon, wie Tensor-Kerne (TCs), Tensor-Verarbeitungseinheiten (TPUs), Pixel-Visual-Cores (PVCs), Vision-Verarbeitungseinheiten (VPUs), Grafikverarbeitungscluster (GPCs), Texturverarbeitungscluster (TPCs), Streaming-Multiprozessoren (SMs), Tree-Traversal-Einheiten (TTUs), Artificial Intelligence Accelerators (AIAs), Deep Learning Accelerators (DLAs), Arithmetik-Logik-Einheiten (ALUs), anwendungsspezifische integrierte Schaltungen (ASICs), Fließkomma-Einheiten (FPUs), Eingabe-/Ausgabe-Elemente (I/O), Peripheral Component Interconnect (PCI)- oder Peripheral Component Interconnect Express (PCIe)-Elemente und/oder dergleichen.

[0249] Die Kommunikationsschnittstelle 1810 kann einen oder mehrere Empfänger, Sender und/oder Transceiver enthalten, die es dem Computergerät 1800 ermöglichen, mit anderen Computergeräten über ein elektronisches Kommunikationsnetzwerk zu kommunizieren, einschließlich drahtgebundener und/oder drahtloser Kommunikation. Die Kommunikationsschnittstelle 1810 kann Komponenten und Funktionen enthalten, die die Kommunikation über eine Reihe verschiedener Netzwerke ermöglichen, wie z. B. drahtlose Netzwerke (z. B. Wi-Fi, Z-Wave, Bluetooth, Bluetooth LE, ZigBee usw.), drahtgebundene Netzwerke (z. B. Kommunikation über Ethernet oder InfiniBand), Weitverkehrsnetzwerke mit geringer Leistung (z. B. LoRaWAN, SigFox usw.) und/oder das Internet.

[0250] Die I/O-Ports 1812 können es ermöglichen, dass das Computergerät 1800 logisch mit anderen Geräten gekoppelt werden kann, einschließlich der I/O-Komponenten 1814, der Präsentationskomponente(n) 1818 und/oder anderer Komponenten, von denen einige in das Computergerät 1800 eingebaut (z. B. integriert) sein können. Illustrative I/O-Komponenten 1814 umfassen ein Mikrofon, eine Maus, eine Tastatur, einen Joystick, ein Gamepad, einen Gamecontroller, eine Satellitenschüssel, einen Scanner, einen Drucker, ein drahtloses Gerät usw. Die I/O-Komponenten 1814 können eine natürliche Benutzerschnittstelle (NUI) bereitstellen, die Luftgesten, Sprache oder andere physiologische Eingaben eines Benutzers verarbeitet. In einigen Fällen können die Eingaben zur weiteren Verarbeitung an ein geeignetes Netzwerkelement übertragen werden. Eine NUI kann eine beliebige Kombination aus Spracherkennung, Stifterkennung, Gesichtserkennung, biometrischer Erkennung, Gestenerkennung sowohl auf dem Bildschirm als auch neben dem Bildschirm, Luftgesten, Kopf- und Augenverfolgung und Berührungserkennung (wie unten ausführlicher beschrieben) in Verbindung mit einer Anzeige des Rechenvorrichtung 1800 implementieren. Die Computervorrichtung 1800 kann Tiefenkameras, wie stereoskopische Kamerasysteme, Infrarotkamerasysteme, RGB-Kamerasysteme, Touchscreen-Technologie und Kombinationen davon, zur Gestenerkennung und -erfassung enthalten. Darüber hinaus kann das Computergerät 1800 Beschleunigungsmesser oder Gyroskope (z. B. als Teil einer Trägheitsmesseinheit (IMU)) enthalten, die die Erkennung von Bewegungen ermöglichen. In einigen Beispielen kann die Ausgabe der Beschleunigungsmesser oder Gyroskope von der Computervorrichtung 1800 verwendet werden, um immersive erweiterte Realität oder virtuelle Realität darzustellen.

[0251] Die Stromversorgung 1816 kann eine festverdrahtete Stromversorgung, eine Batteriestromversorgung oder eine Kombination davon sein. Die Stromversorgung 1816 kann das Computergerät 1800 mit Strom versorgen, um den Betrieb der Komponenten des Rechenvorrichtung 1800 zu ermöglichen.

[0252] Die Präsentationskomponente(n) 1818 können eine Anzeige (z. B. einen Monitor, einen Touchscreen, einen Fernsehbildschirm, ein Heads-up-Display (HUD), andere Anzeigetypen oder eine Kombination davon), Lautsprecher und/oder andere Präsentationskomponenten umfassen. Die Präsentationskomponente(n) 1818 können Daten von anderen Komponenten (z. B. der/den GPU(s) 1808, der/den CPU(s) 1806 usw.) empfangen und die Daten ausgeben (z. B. als Bild, Video, Ton usw.).

Beispielhaftes Datenzentrum

[0253] **Fig. 19** zeigt ein Beispiel für ein Datenzentrum 1900, das in mindestens einer Ausführungsform der vorliegenden Offenbarung verwendet werden kann. Das Datenzentrum 1900 kann eine Datenzentrum-Infrastrukturschicht 1910, eine Framework-Schicht 1920, eine Softwareschicht 1930 und/oder eine Anwendungsschicht 1940 umfassen.

[0254] Wie in **Fig. 19** dargestellt, kann die Datenzentrum-Infrastrukturschicht 1910 einen Ressourcen-Orchestrator 1912, gruppierte Rechenressourcen 1914 und Knotenrechenressourcen („Knoten-C.R.s“) 1916(1)-1916(N) umfassen, wobei „N“ eine beliebige ganze, positive Zahl darstellt. In mindestens einer Ausführungsform können die Knoten-C.R.s 1916(1)-1916(N) eine beliebige Anzahl von Zentraleinheiten („CPUs“) oder anderen Prozessoren (einschließlich Beschleunigern, feldprogrammierbaren Gate-Arrays (FPGAs), Grafikprozessoren oder Grafikverarbeitungseinheiten (GPUs) usw.), Speichervorrichtungen (z. B., dynamischer Festwertspeicher), Speichergeräte (z. B. Festkörper- oder Plattenlaufwerke), Netzwerk-Eingabe-/Ausgabegeräte („NW I/O“), Netzwerk-Switches, virtuelle Maschinen („VMs“), Stromversorgungsmodule und/oder Kühlmodule usw. In einigen Ausführungsformen kön-

nen ein oder mehrere Knoten-C.R.s unter den Knoten-C.R.s 1916(1)-1916(N) einem Server entsprechen, der über eine oder mehrere der oben erwähnten Rechenressourcen verfügt. Darüber hinaus können in einigen Ausführungsformen die Knoten C.R.s 1916(1)-19161(N) eine oder mehrere virtuelle Komponenten enthalten, wie z. B. vGPUs, vCPUs und/oder dergleichen, und/oder einer oder mehrere der Knoten C.R.s 1916(1)-1916(N) können einer virtuellen Maschine (VM) entsprechen.

[0255] In mindestens einer Ausführungsform können die gruppierten Rechenressourcen 1914 separate Gruppierungen von Knoten-CRs 1916 umfassen, die in einem oder mehreren Racks (nicht dargestellt) oder in vielen Racks in Datenzentren an verschiedenen geografischen Standorten (ebenfalls nicht dargestellt) untergebracht sind. Getrennte Gruppierungen von Knoten-C.R.s 1916 innerhalb gruppierter Rechenressourcen 1914 können gruppierte Rechen-, Netzwerk-, Speicher- oder Speicherressourcen umfassen, die zur Unterstützung einer oder mehrerer Arbeitslasten konfiguriert oder zugewiesen werden können. In mindestens einer Ausführungsform können mehrere Knoten-CRs 1916 einschließlich CPUs, GPUs und/oder anderer Prozessoren in einem oder mehreren Racks gruppiert werden, um Rechenressourcen zur Unterstützung einer oder mehrerer Arbeitslasten bereitzustellen. Das eine oder die mehreren Racks können auch eine beliebige Anzahl von Stromversorgungsmodulen, Kühlmodulen und/oder Netzwerk-Switches in beliebiger Kombination enthalten.

[0256] Der Ressourcen-Orchestrator 1922 kann einen oder mehrere Knoten-CRs 1916(1)-1916(N) und/oder gruppierte Rechenressourcen 1914 konfigurieren oder anderweitig steuern. In mindestens einer Ausführungsform kann der Ressourcen-Orchestrator 1922 eine Software-Design-Infrastruktur („SDI“)-Verwaltungseinheit für das Datenzentrum 1900 umfassen. Der Ressourcen-Orchestrator 1922 kann Hardware, Software oder eine Kombination davon umfassen.

[0257] In mindestens einer Ausführungsform, wie in **Fig. 19** gezeigt, kann die Framework-Schicht 1920 einen Job-Scheduler 1932, einen Konfigurationsmanager 1934, einen Ressourcenmanager 1936 und/oder ein verteiltes Dateisystem 1938 enthalten. Die Framework-Schicht 1920 kann ein Framework zur Unterstützung der Software 1932 der Softwareschicht 1930 und/oder einer oder mehrerer Anwendungen 1942 der Anwendungsschicht 1940 enthalten. Die Software 1932 oder die Anwendung(en) 1942 können jeweils webbasierte Dienstsoftware oder - anwendungen umfassen, wie sie von Amazon Web Services, Google Cloud und Microsoft Azure bereitgestellt werden. Bei der Framework-Schicht 1920 kann es sich um eine Art freies und quelloffenes Software-Webanwendungs-Framework wie Apache Spark™ (im Folgenden „Spark“) handeln, das das verteilte Dateisystem 1938 für die Verarbeitung großer Datenmengen (z. B. „Big Data“) nutzen kann, ohne darauf beschränkt zu sein. In mindestens einer Ausführungsform kann der Job-Scheduler 1932 einen Spark-Treiber enthalten, um die Planung von Arbeitslasten zu erleichtern, die von verschiedenen Schichten des Datenzentrums 1900 unterstützt werden. Der Konfigurationsmanager 1934 kann in der Lage sein, verschiedene Schichten zu konfigurieren, wie z. B. die Softwareschicht 1930 und die Framework-Schicht 1920 einschließlich Spark und das verteilte Dateisystem 1938 zur Unterstützung der Verarbeitung großer Datenmengen. Der Ressourcenmanager 1936 kann in der Lage sein, geclusterte oder gruppierte Rechenressourcen zu verwalten, die zur Unterstützung des verteilten Dateisystems 1938 und des Job-Schedulers 1932 zugeordnet sind. In mindestens einer Ausführungsform können geclusterte oder gruppierte Rechenressourcen gruppierte Rechenressourcen 1914 auf der Infrastrukturschicht 1910 des Datenzentrums umfassen. Der Ressourcenmanager 1036 kann sich mit dem Ressourcen-Orchestrator 1912 abstimmen, um diese zugeordneten oder zugewiesenen Computerressourcen zu verwalten.

[0258] In mindestens einer Ausführungsform kann die in der Softwareschicht 1930 enthaltene Software 1932 Software enthalten, die von mindestens Teilen der Knoten C.R.s 1916(1)-1916(N), der gruppierten Rechenressourcen 1914 und/oder des verteilten Dateisystems 1938 der Framework-Schicht 1920 verwendet wird. Eine oder mehrere Arten von Software können unter anderem Internet-Such-Software, E-Mail-Viren-Scan-Software, Datenbank-Software und Streaming-Video-Content-Software umfassen.

[0259] In mindestens einer Ausführungsform können die in der Anwendungsschicht 1940 enthaltene (n) Anwendung(en) 1942 eine oder mehrere Arten von Anwendungen umfassen, die von mindestens Teilen der Knoten C.R.s 1916(1)-1916(N), gruppierten Rechenressourcen 1914 und/oder dem verteilten Dateisystem 1938 der Framework-Schicht 1920 verwendet werden. Eine oder mehrere Arten von Anwendungen können eine beliebige Anzahl von Genomanwendungen, kognitiven Berechnungen und maschinellen Lernanwendungen, einschließlich Trainings- oder Inferenzsoftware, maschinelle Lernsoftware (z. B. PyTorch, TensorFlow, Caffe usw.) und/oder andere maschinelle Lernanwendungen, die in Verbindung mit einer oder mehreren Ausführungsformen verwendet werden, umfassen, sind aber nicht darauf beschränkt.

[0260] In mindestens einer Ausführungsform können der Konfigurationsmanager 1934, der Ressourcenmanager 1936 und der Ressourcen-Orchestrator

1912 eine beliebige Anzahl und Art von selbstmodifizierenden Aktionen implementieren, die auf einer beliebigen Menge und Art von Daten basieren, die in einer technisch machbaren Weise erfasst wurden. Selbstmodifizierende Aktionen können einen Datenzentrumsbetreiber des Datenzentrums 1900 davon entlasten, möglicherweise schlechte Konfigurationsentscheidungen zu treffen und möglicherweise nicht ausgelastete und/oder schlecht funktionierende Teile eines Datenzentrums zu vermeiden.

[0261] Das Datenzentrum 1900 kann Werkzeuge, Dienste, Software oder andere Ressourcen enthalten, um ein oder mehrere maschinelle Lernmodelle zu trainieren oder Informationen unter Verwendung eines oder mehrerer maschineller Lernmodelle gemäß einer oder mehrerer hier beschriebener Ausführungsformen vorherzusagen oder abzuleiten. Beispielsweise kann ein maschinelles Lernmodell oder können maschinelle Lernmodelle trainiert werden, indem Gewichtsparameter gemäß einer neuronalen Netzwerkarchitektur unter Verwendung von Software und/oder Rechenressourcen berechnet werden, die oben in Bezug auf das Datenzentrum 1900 beschrieben wurden. In mindestens einer Ausführungsform können trainierte oder eingesetzte maschinelle Lernmodelle, die einem oder mehreren neuronalen Netzen entsprechen, verwendet werden, um Informationen abzuleiten oder vorherzusagen, wobei die oben beschriebenen Ressourcen in Bezug auf das Datenzentrum 1900 verwendet werden, indem Gewichtungsparameter verwendet werden, die durch eine oder mehrere Trainingstechniken berechnet werden, wie z. B., aber nicht beschränkt auf die hier beschriebenen.

[0262] In mindestens einer Ausführungsform kann das Datenzentrum 1900 CPUs, anwendungsspezifische integrierte Schaltungen (ASICs), GPUs, FPGAs und/oder andere Hardware (oder entsprechende virtuelle Rechenressourcen) verwenden, um Training und/oder Inferencing unter Verwendung der oben beschriebenen Ressourcen durchzuführen. Darüber hinaus können eine oder mehrere der oben beschriebenen Software- und/oder Hardwareressourcen als Dienst konfiguriert werden, um Benutzern das Training oder die Inferenzierung von Informationen zu ermöglichen, wie z. B. Bilderkennung, Spracherkennung oder andere Dienste der künstlichen Intelligenz.

Beispielhafte Netzwerkumgebungen

[0263] Netzwerkumgebungen, die zur Verwendung bei der Implementierung von Ausführungsformen der Offenbarung geeignet sind, können ein oder mehrere Client-Geräte, Server, Network Attached Storage (NAS), andere Backend-Geräte und/oder andere Gerätetypen umfassen. Die Client-Geräte, Server und/oder anderen Gerätetypen (z. B. jedes Gerät) können auf einer oder mehreren Instanzen des/der Rechengeräts/e 1800 der **Fig. 18** implementiert werden - z. B. kann jedes Gerät ähnliche Komponenten, Merkmale und/oder Funktionalität des/der Rechengeräts/e 1800 enthalten. Wenn Backend-Geräte (z. B. Server, NAS usw.) implementiert sind, können die Backend-Geräte außerdem Teil eines Datenzentrums 1900 sein, dessen Beispiel hier in Bezug auf **Fig. 19** näher beschrieben wird.

[0264] Die Komponenten einer Netzwerkumgebung können über ein oder mehrere Netzwerke miteinander kommunizieren, die drahtgebunden, drahtlos oder beides sein können. Das Netz kann mehrere Netze oder ein Netz von Netzen umfassen. Beispielsweise kann das Netzwerk ein oder mehrere Wide Area Networks (WANs), ein oder mehrere Local Area Networks (LANs), ein oder mehrere öffentliche Netzwerke wie das Internet und/oder ein öffentliches Telefonnetz (PSTN) und/oder ein oder mehrere private Netzwerke umfassen. Wenn das Netz ein drahtloses Telekommunikationsnetz umfasst, können Komponenten wie eine Basisstation, ein Kommunikationsturm oder sogar Zugangspunkte (sowie andere Komponenten) eine drahtlose Verbindung ermöglichen.

[0265] Zu den kompatiblen Netzwerkumgebungen gehören eine oder mehrere Peer-to-Peer-Netzwerkumgebungen - in diesem Fall kann ein Server nicht in einer Netzwerkumgebung enthalten sein - und eine oder mehrere Client-Server-Netzwerkumgebungen - in diesem Fall können ein oder mehrere Server in einer Netzwerkumgebung enthalten sein. In Peer-to-Peer-Netzwerkumgebungen kann die hier beschriebene Funktionalität in Bezug auf einen oder mehrere Server auf einer beliebigen Anzahl von Client-Geräten implementiert werden.

[0266] In mindestens einer Ausführungsform kann eine Netzumgebung eine oder mehrere Cloud-basierte Netzumgebungen, eine verteilte Rechenumgebung, eine Kombination davon usw. umfassen. Eine Cloud-basierte Netzwerkumgebung kann eine Framework-Schicht, einen Job-Scheduler, einen Ressourcenmanager und ein verteiltes Dateisystem umfassen, die auf einem oder mehreren Servern implementiert sind, die einen oder mehrere Kernnetzwerkserver und/oder Edge-Server umfassen können. Eine Framework-Schicht kann ein Framework zur Unterstützung von Software einer Softwareschicht und/oder einer oder mehrerer Anwendungen einer Anwendungsschicht umfassen. Die Software oder die Anwendung(en) können jeweils webbasierte Dienstsoftware oder Anwendungen umfassen. In Ausführungsformen können ein oder mehrere Client-Geräte die webbasierte Dienstsoftware oder Anwendungen nutzen (z. B. durch Zugriff auf die Dienstsoftware und/oder Anwendungen über eine oder mehrere Anwendungsprogrammierschnittstel-

len (APIs)). Bei der Framework-Schicht kann es sich um eine Art von freiem und quelloffenem Software-Webanwendungs-Framework handeln, das z. B. ein verteiltes Dateisystem für die Verarbeitung großer Datenmengen (z. B. „Big Data“) verwendet, ohne darauf beschränkt zu sein.

[0267] Eine Cloud-basierte Netzwerkumgebung kann Cloud-Computing und/oder Cloud-Speicher bereitstellen, die eine beliebige Kombination der hier beschriebenen Rechen- und/oder Datenspeicherfunktionen (oder einen oder mehrere Teile davon) ausführen. Jede dieser verschiedenen Funktionen kann über mehrere Standorte von zentralen oder Kernservern (z. B. von einem oder mehreren Datenzentren, die über einen Staat, eine Region, ein Land, den Globus usw. verteilt sein können) verteilt sein. Befindet sich eine Verbindung zu einem Benutzer (z. B. einem Client-Gerät) relativ nahe an einem oder mehreren Edge-Servern, kann ein Kernserver mindestens einen Teil der Funktionalität dem oder den Edge-Servern zuweisen. Eine Cloud-basierte Netzwerkumgebung kann privat (z. B. auf eine einzelne Organisation beschränkt), öffentlich (z. B. für viele Organisationen verfügbar) und/oder eine Kombination davon (z. B. eine hybride Cloud-Umgebung) sein.

[0268] Das (die) Client-Gerät(e) können mindestens einige der Komponenten, Merkmale und Funktionen des (der) hier in Bezug auf **Fig. 18** beschriebenen Beispiel-Rechenvorrichtung (-geräte) 1800 enthalten. Beispielhaft und ohne Einschränkung kann ein Client-Gerät als Personal Computer (PC), Laptop-Computer, mobiles Gerät, Smartphone, Tablet-Computer, intelligente Uhr, tragbarer Computer, persönlicher digitaler Assistent (PDA), MP3-Player, Virtual-Reality-Headset, Global Positioning System (GPS) oder -Gerät, Videoplayer, Videokamera, Überwachungsgerät oder -system, Fahrzeug ein Boot, ein fliegendes Schiff, eine virtuelle Maschine, eine Drohne, ein Roboter, ein tragbares Kommunikationsgerät, ein Krankenhausgerät, ein Spielgerät oder -system, ein Unterhaltungssystem, ein Fahrzeugcomputersystem, ein eingebettetes Systemcontroller, eine Fernbedienung, ein Gerät, ein Unterhaltungselektronikgerät, eine Workstation, ein Edge-Gerät, eine beliebige Kombination dieser beschriebenen Geräte oder jedes andere geeignete Gerät.

[0269] Die Offenbarung kann im allgemeinen Kontext von Computercode oder maschinell verwendbaren Anweisungen, einschließlich von computerausführbaren Anweisungen wie Programmmodulen, beschrieben werden, die von einem Computer oder einer anderen Maschine, z. B. einem persönlichen Datenassistenten oder einem anderen Handheld-Gerät, ausgeführt werden. Im Allgemeinen beziehen sich Programmmodule, einschließlich Routinen, Programme, Objekte, Komponenten, Datenstrukturen usw., auf Code, der bestimmte Aufgaben ausführt oder bestimmte abstrakte Datentypen implementiert. Die Offenbarung kann in einer Vielzahl von Systemkonfigurationen angewendet werden, einschließlich Handheld-Geräten, Unterhaltungselektronik, Allzweckcomputern, spezielleren Datenverarbeitungsgeräten usw. Die Offenbarung kann auch in verteilten Computerumgebungen angewendet werden, in denen Aufgaben von ferngesteuerten Geräten ausgeführt werden, die über ein Kommunikationsnetz verbunden sind.

[0270] Wenn hier von „und/oder“ in Bezug auf zwei oder mehr Elemente die Rede ist, sollte dies so verstanden werden, dass nur ein Element oder eine Kombination von Elementen gemeint ist. So kann beispielsweise „Element A, Element B und/oder Element C“ nur Element A, nur Element B, nur Element C, Element A und Element B, Element A und Element C, Element B und Element C oder die Elemente A, B und C umfassen. Darüber hinaus kann „mindestens eines der Elemente A oder B“ mindestens eines der Elemente A, mindestens eines der Elemente B oder mindestens eines der Elemente A und mindestens eines der Elemente B einschließen.

[0271] Der Gegenstand der vorliegenden Offenbarung wird hier mit einer Genauigkeit beschrieben, die den gesetzlichen Anforderungen entspricht. Die Beschreibung selbst soll jedoch den Umfang dieser Offenbarung nicht einschränken. Vielmehr haben die Erfinder in Betracht gezogen, dass der beanspruchte Gegenstand auch auf andere Weise verkörpert werden könnte, um verschiedene Schritte oder Kombinationen von Schritten, die den in diesem Dokument beschriebenen ähnlich sind, in Verbindung mit anderen gegenwärtigen oder zukünftigen Technologien zu umfassen. Obwohl die Begriffe „Schritt“ und/oder „Block“ hier verwendet werden können, um verschiedene Elemente der angewandten Methoden zu bezeichnen, sollten die Begriffe nicht so ausgelegt werden, dass sie eine bestimmte Reihenfolge unter oder zwischen den verschiedenen hier offenbarten Schritten implizieren, es sei denn, die Reihenfolge der einzelnen Schritte wird ausdrücklich beschrieben.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Patentliteratur

- US 16818551 [0092]
- US 17/187350 [0107]
- US 62/514404 [0108]
- US 16/514404 [0108]
- US 16101232 [0174]

Zitierte Nicht-Patentliteratur

- ISO 26262 [0173]

Patentansprüche

1. Verfahren, das aufweist:
Erzeugen, mindestens basierend auf Bilddaten, die während einer Erfassungssitzung in einer Umgebung erfasst werden, einer ersten Darstellung einer dreidimensionalen, 3D, Oberflächenstruktur einer Komponente der Umgebung;
Erzeugen, mindestens basierend auf LiDAR-Daten, die mit den Bilddaten verbunden sind und während der Erfassungssitzung erfasst werden, einer zweiten Darstellung der 3D-Oberflächenstruktur der Komponente; und
Trainieren eines oder mehrerer neuronaler Netze, NNs, um eine verdichtete Darstellung der 3D-Oberflächenstruktur unter Verwendung der ersten Darstellung der 3D-Oberflächenstruktur als Eingabetrainingsdaten und unter Verwendung der zweiten Darstellung der 3D-Oberflächenstruktur als Ground-Truth-Trainingsdaten zu erzeugen.

2. Verfahren nach Anspruch 1, wobei das Erzeugen der ersten Darstellung der 3D-Oberflächenstruktur das Anwenden einer 3D-Oberflächenstrukturschätzung auf die Bilddaten aufweist.

3. Verfahren nach Anspruch 1 oder 2, wobei das Erzeugen der ersten Darstellung der 3D-Oberflächenstruktur aufweist:
Identifizieren eines ausgerichteten Satzes der Bilddaten und der LiDAR-Daten; und
Erzeugen der zweiten Darstellung der 3D-Oberflächenstruktur mindestens basierend auf dem ausgerichteten Satz der Bilddaten und der LiDAR-Daten.

4. Verfahren nach einem der vorhergehenden Ansprüche, wobei das Erzeugen der ersten Darstellung der 3D-Oberflächenstruktur das Auffüllen eines oder mehrerer fehlender Werte der LiDAR-Daten unter Verwendung einer Delaunay-Triangulierung aufweist.

5. Verfahren nach einem der vorhergehenden Ansprüche, wobei das Erzeugen der ersten Darstellung der 3D-Oberflächenstruktur das Akkumulieren der LiDAR-Daten von mehreren LiDAR-Sensoren aufweist.

6. Verfahren nach einem der vorhergehenden Ansprüche, wobei das Erzeugen der zweiten Darstellung der 3D-Oberflächenstruktur aufweist:
durch ein Kennzeichnungswerkzeug Erhalten von einer oder mehreren Eingaben, die annotierte Punkte der LiDAR-Daten identifizieren, die der Komponente der Umgebung entsprechen, während ein ausgerichteter Satz der Bilddaten und der LiDAR-Daten präsentiert wird; und
Erzeugen der zweiten Darstellung der 3D-Oberflächenstruktur der Komponente mindestens basierend auf der Projektion der annotierten Punkte, um ein Projektionsbild zu erzeugen.

7. Verfahren nach einem der vorhergehenden Ansprüche, das ferner das Anwenden einer Objekterkennung auf die Bilddaten aufweist, um eine Ground-Truth-Segmentierungsmaske zu erzeugen, die die Komponente der Umgebung abgrenzt.

8. Verfahren nach einem der vorhergehenden Ansprüche, wobei die zweite Darstellung der 3D-Oberflächenstruktur der Komponente eine zweidimensionale, 2D, Ground-Truth-Höhenkarte aufweist, die einen oder mehrere Höhenwerte der Komponente repräsentiert, wobei das Trainieren des einen oder der mehreren NNs das Vergleichen einer vorhergesagten 2D-Höhenkarte mit der 2D-Ground-Truth-Höhenkarte und das Anwenden einer Ground-Truth-Segmentierungsmaske aufweist, die die Komponente abgrenzt, um eine oder mehrere Aktualisierungen des einen oder der mehreren NNs basierend auf einem oder mehreren Werten der vorhergesagten 2D-Höhenkarte aufzuheben, die nicht der Komponente entsprechen.

9. Verfahren nach einem der vorhergehenden Ansprüche, wobei das eine oder die mehreren NNs einen ersten Ausgangskanal, der einen oder mehrere Höhenwerte der Komponente regressiert, und einen zweiten Ausgangskanal aufweisen, der einen oder mehrere Konfidenzwerte regressiert, die dem einen oder den mehreren Höhenwerten entsprechen, wobei das Trainieren des einen oder der mehreren NNs das Anwenden einer Verlustfunktion aufweist, wobei die Verlustfunktion vorhergesagte und Ground-Truth-Höhenwerte vergleicht und auf vorhergesagten Konfidenzwerten basiert.

10. Verfahren nach einem der vorhergehenden Ansprüche, das ferner das Verwenden des einen oder der mehreren NNs aufweist, um mindestens eines auszuführen von:
Anpassen des Aufhängungssystems eines Ego-Objekts,
Navigieren des Ego-Objekts, um einer erkannten Protuberanz auszuweichen, oder
Anwenden einer Beschleunigung oder Abbremsung auf das Ego-Objekt mindestens basierend auf einer erkannten Oberflächenneigung.

11. Prozessor, der eine oder mehrere Schaltungen aufweist, um:
auf Bilddaten und LiDAR-Daten zuzugreifen, die während einer Erfassungssitzung in einer Umgebung erfasst werden;
mindestens basierend auf den Bilddaten eine unvollständige Darstellung einer dreidimensionalen, 3D, Oberflächenstruktur der Straße in der Umgebung zu erzeugen;
mindestens basierend auf einer Kennzeichnung der

LiDAR-Daten eine zweite Darstellung der 3D-Oberflächenstruktur der Straße zu erzeugen; und
ein oder mehrere neuronale Netze, NNs, zu trainieren, um unter Verwendung der unvollständigen Darstellung der 3D-Oberflächenstruktur als Eingabetrainingsdaten und unter Verwendung der zweiten Darstellung der 3D-Oberflächenstruktur als Ground-Truth-Trainingsdaten eine verdichtete Darstellung der 3D-Oberflächenstruktur zu erzeugen.

12. Prozessor nach Anspruch 11, wobei die eine oder die mehreren Schaltungen ferner dazu dienen, die unvollständige Darstellung der 3D-Oberflächenstruktur mindestens basierend auf der Anwendung der 3D-Oberflächenstrukturschätzung auf die Bilddaten zu erzeugen.

13. Prozessor nach Anspruch 11 oder 12, wobei die eine oder die mehreren Schaltungen ferner dazu dienen, die unvollständige Darstellung der 3D-Oberflächenstruktur mindestens basierend auf dem Auffüllen eines oder mehrerer fehlender Werte der LiDAR-Daten unter Verwendung einer Delaunay-Triangulierung zu erzeugen.

14. Prozessor nach einem der Ansprüche 11 bis 13, wobei die eine oder die mehreren Schaltungen ferner dazu dienen, die unvollständige Darstellung der 3D-Oberflächenstruktur mindestens basierend auf der Akkumulation der LiDAR-Daten von mehreren LiDAR-Sensoren zu erzeugen.

15. Prozessor nach einem der Ansprüche 11 bis 14, wobei die eine oder die mehreren Schaltungen ferner dazu dienen, die zweite Darstellung der 3D-Oberflächenstruktur zu erzeugen mindestens basierend auf:
durch ein Kennzeichnungswerkzeug, Erhalten einer oder mehrerer Eingaben, die einen oder mehrere annotierte Punkte der LiDAR-Daten identifizieren, die der Straße entsprechen, während ein ausgerichteter Satz der Bilddaten und der LiDAR-Daten präsentiert wird; und
Erzeugen der zweiten Darstellung der 3D-Oberflächenstruktur der Straße mindestens basierend auf Projizieren des einen oder der mehreren annotierten Punkte, um ein Projektionsbild zu erzeugen.

16. Prozessor nach einem der Ansprüche 11 bis 15, wobei die zweite Darstellung der 3D-Oberflächenstruktur der Straße eine zweidimensionale, 2D, Ground-Truth-Höhenkarte aufweist, die einen oder mehrere Höhenwerte der Straße repräsentiert, wobei die eine oder die mehreren Schaltungen ferner dazu dienen, das eine oder die mehreren NNs mindestens basierend auf Vergleichen einer vorhergesagten 2D-Höhenkarte mit der 2D-Ground-Truth-Höhenkarte und Anwenden einer Ground-Truth-Segmentierungsmaske zu trainieren, die die Straße abgrenzt, um eine oder mehrere Aktualisierungen des einen oder der mehreren NNs basierend auf einem oder mehreren Werten der vorhergesagten 2D-Höhenkarte aufzuheben, die nicht der Straße entsprechen.

17. Prozessor nach einem der Ansprüche 11 bis 16, wobei das eine oder die mehreren NNs einen ersten Ausgangskanal, der einen oder mehrere Höhenwerte der Straße regressiert, und einen zweiten Ausgangskanal aufweisen, der einen oder mehrere Konfidenzwerte regressiert, die dem einen oder den mehreren Höhenwerten entsprechen, wobei die eine oder die mehreren Schaltungen ferner dazu dienen, das eine oder die mehreren NNs mindestens basierend auf Anwenden einer Verlustfunktion zu trainieren, wobei die Verlustfunktion vorhergesagte und Ground-Truth-Höhenwerte vergleicht und auf vorhergesagten Konfidenzwerten basiert.

18. System, das aufweist:
eine oder mehrere Verarbeitungseinheiten; und
eine oder mehrere Speichereinheiten, die Anweisungen speichern, die, wenn sie von der einen oder den mehreren Verarbeitungseinheiten ausgeführt werden, die eine oder die mehreren Verarbeitungseinheiten veranlassen, Operationen auszuführen, die aufweisen:
Erfassen von Bilddaten und LiDAR-Daten während einer Erfassungssitzung in einer Umgebung;
Erzeugen, mindestens basierend auf Anwenden einer 3D-Strukturabschätzung auf die Bilddaten, einer unvollständigen Darstellung einer dreidimensionalen (3D) Oberflächenstruktur einer Oberfläche in der Umgebung;
Erzeugen, mindestens basierend auf Kennzeichnen der LiDAR-Daten, einer Ground-Truth-Darstellung der 3D-Oberflächenstruktur der Oberfläche; und
Einschließen der unvollständigen Darstellung und der Ground-Truth-Darstellung der 3D-Oberflächenstruktur in einen Trainingsdatensatz.

19. System nach Anspruch 18, wobei das Erzeugen der unvollständigen Darstellung der 3D-Oberflächenstruktur das Anwenden der 3D-Oberflächenstrukturschätzung auf die Bilddaten aufweist.

20. System nach Anspruch 18 oder 19, wobei das Erzeugen der Ground-Truth-Darstellung der 3D-Oberflächenstruktur aufweist:
durch ein Kennzeichnungswerkzeug, Erhalten von einer oder mehreren Eingaben, die einen oder mehrere annotierte Punkte der LiDAR-Daten identifizieren, die der Oberfläche in der Umgebung entsprechen, während ein ausgerichteter Satz der Bilddaten und der LiDAR-Daten präsentiert wird; und
Erzeugen der Ground-Truth-Darstellung der 3D-Oberflächenstruktur der Oberfläche mindestens basierend auf dem Projiziere des einen oder der

mehreren annotierten Punkte, um ein Projektionsbild zu erzeugen.

21. System nach einem der Ansprüche 18 bis 20, wobei das System aus mindestens einem besteht von:
einem Steuerungssystem für eine autonome oder halbautonome Maschine;
einem Wahrnehmungssystem für eine autonome oder halbautonome Maschine;
einem System zum Durchführen von Simulationsoperationen;
einem System zum Durchführen von Deep-Learning-Operationen;
einem System, das unter Verwendung eines Edge-Geräts implementiert wird;
einem System, das unter Verwendung eines Roboters implementiert wird;
einem System, das eine oder mehrere virtuelle Maschinen (VMs) enthält;
einem System, das mindestens teilweise in einem Datenzentrum implementiert wird; oder
einem System, das mindestens teilweise unter Verwendung von Cloud-Computing-Ressourcen implementiert wird.

Es folgen 22 Seiten Zeichnungen

Anhängende Zeichnungen

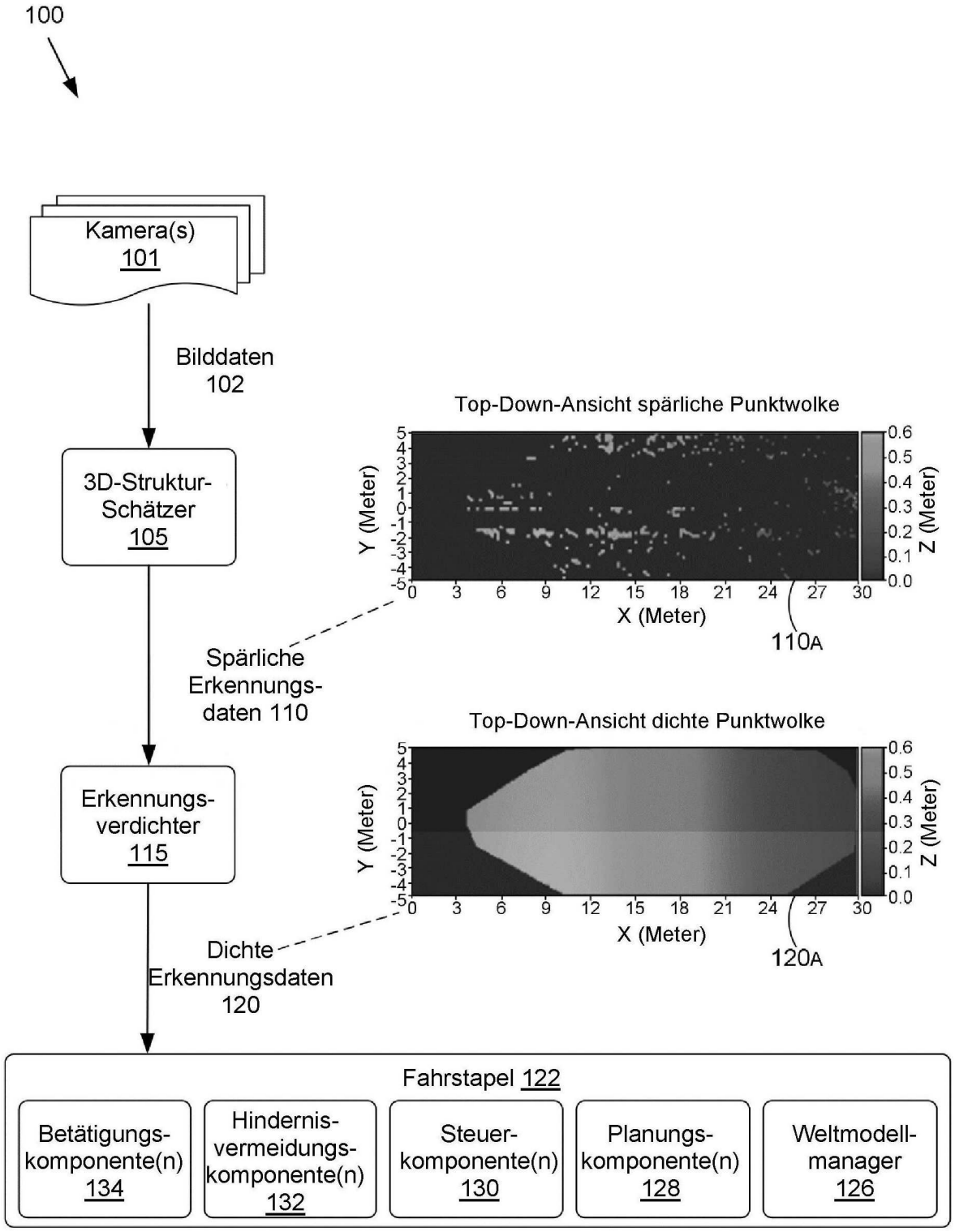


Fig. 1

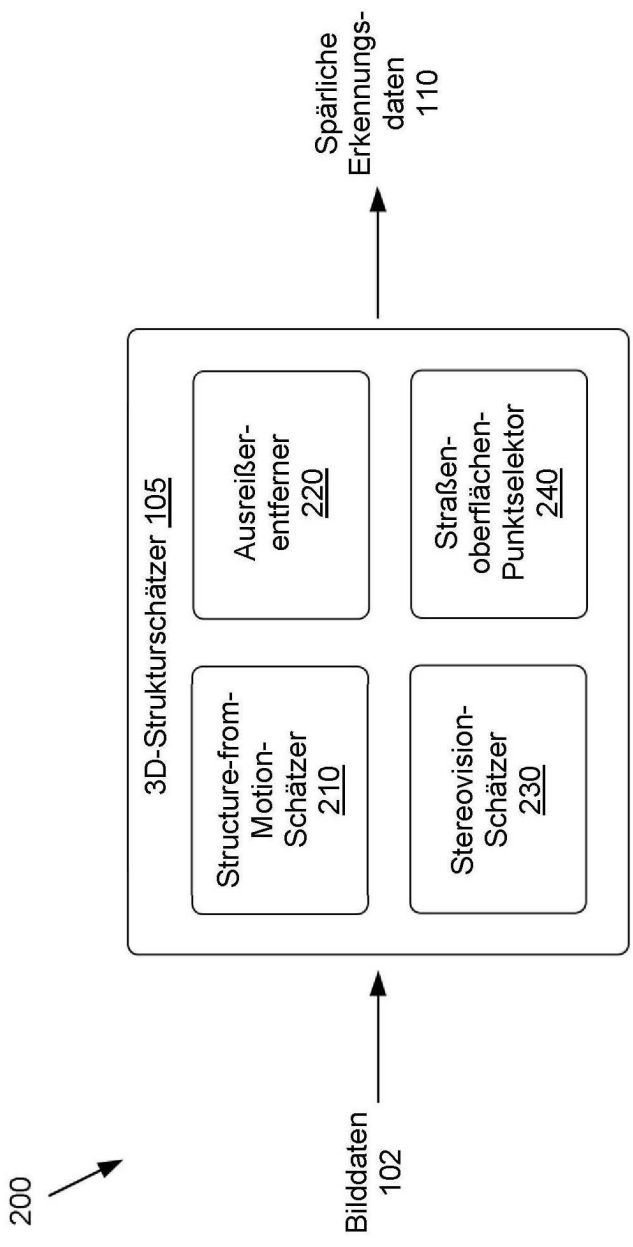

200
Bilddaten
102
3D-Strukturschätzer 105
Structure-from-Motion-Schätzer 210
Ausreißer-entferner 220
Stereovision-Schätzer 230
Straßen-oberflächen-Punktselektor 240
Spärliche Erkennungs-daten 110

Fig. 2

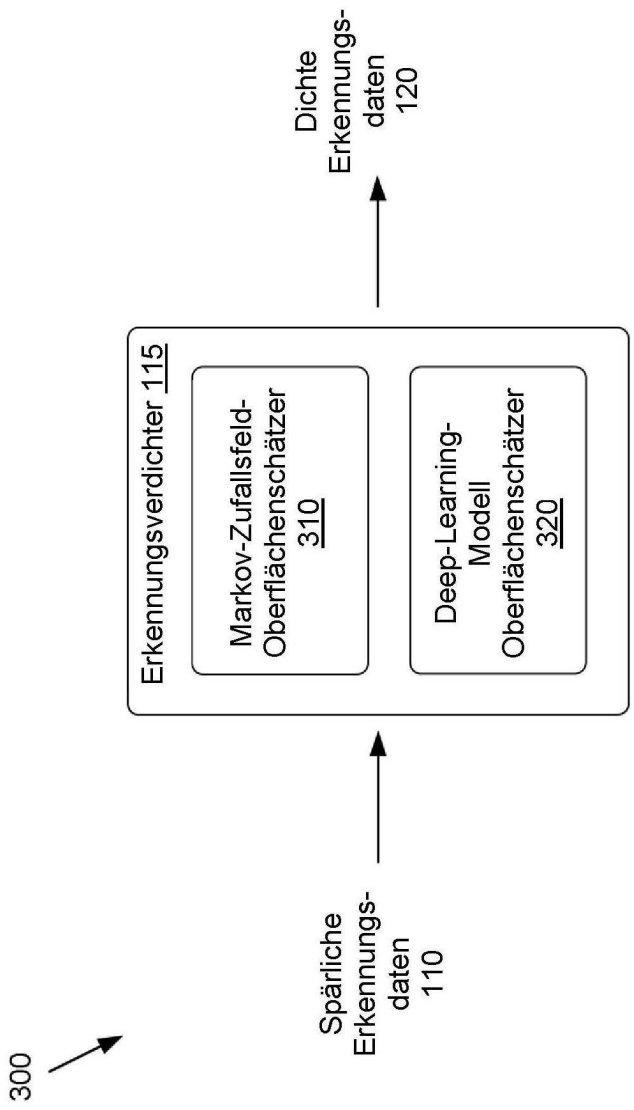

300
Spärliche Erkennungs-daten 110
Erkennungsverdichter 115
Markov-Zufallsfeld-Oberflächenschätzer 310
Deep-Learning-Modell Oberflächenschätzer 320
Dichte Erkennungs-daten 120

Fig. 3

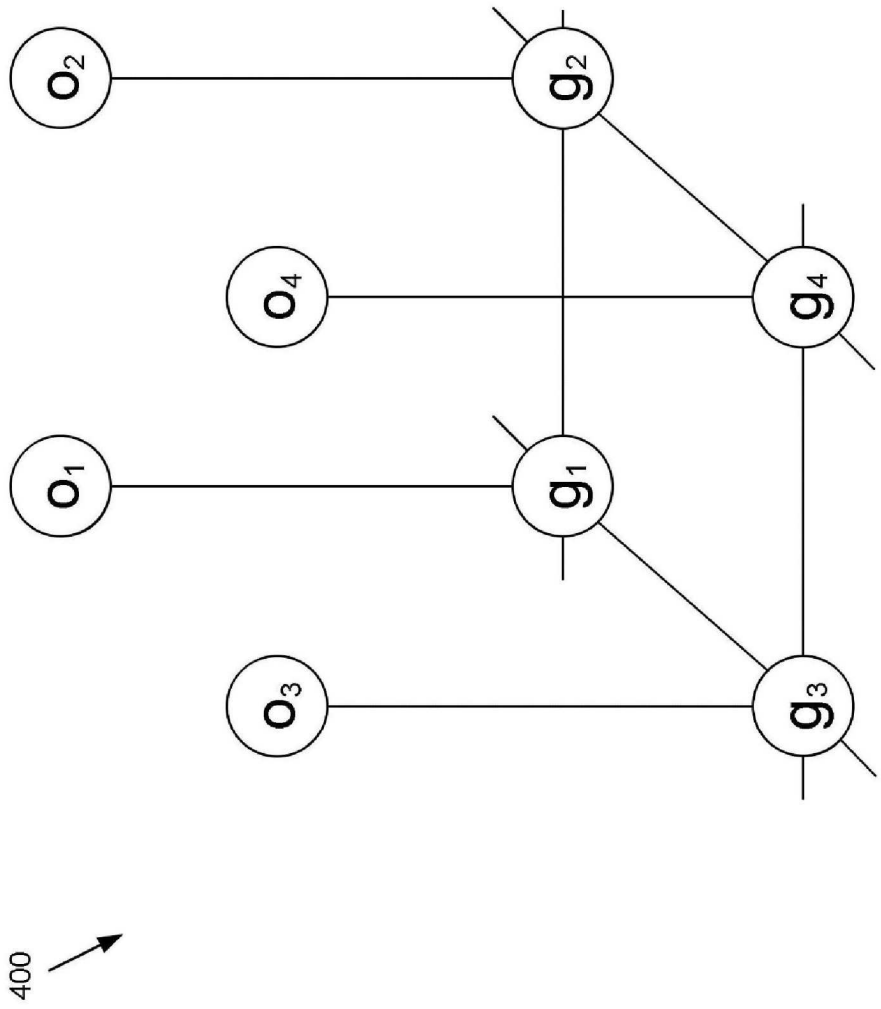


Fig. 4

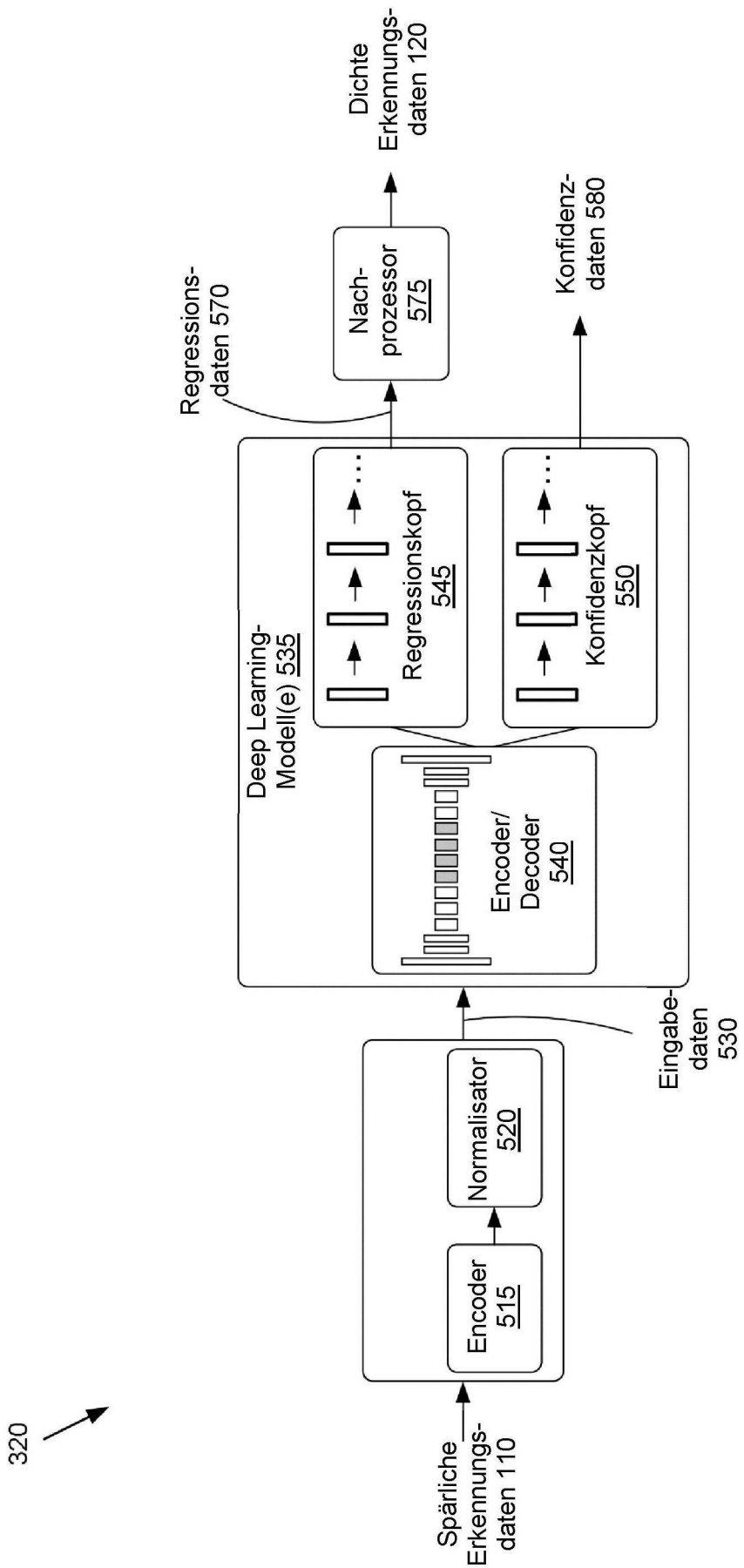

320
Spärliche Erkennungs-daten 110
Encoder 515
Normalisator 520
Eingabe-daten 530
Deep Learning-Modell(e) 535
Encoder/ Decoder 540
Regressionskopf 545
Konfidenzkopf 550
Nach-prozessor 575
Regressions-daten 570
Konfidenz-daten 580
Dichte Erkennungs-daten 120

Fig. 5

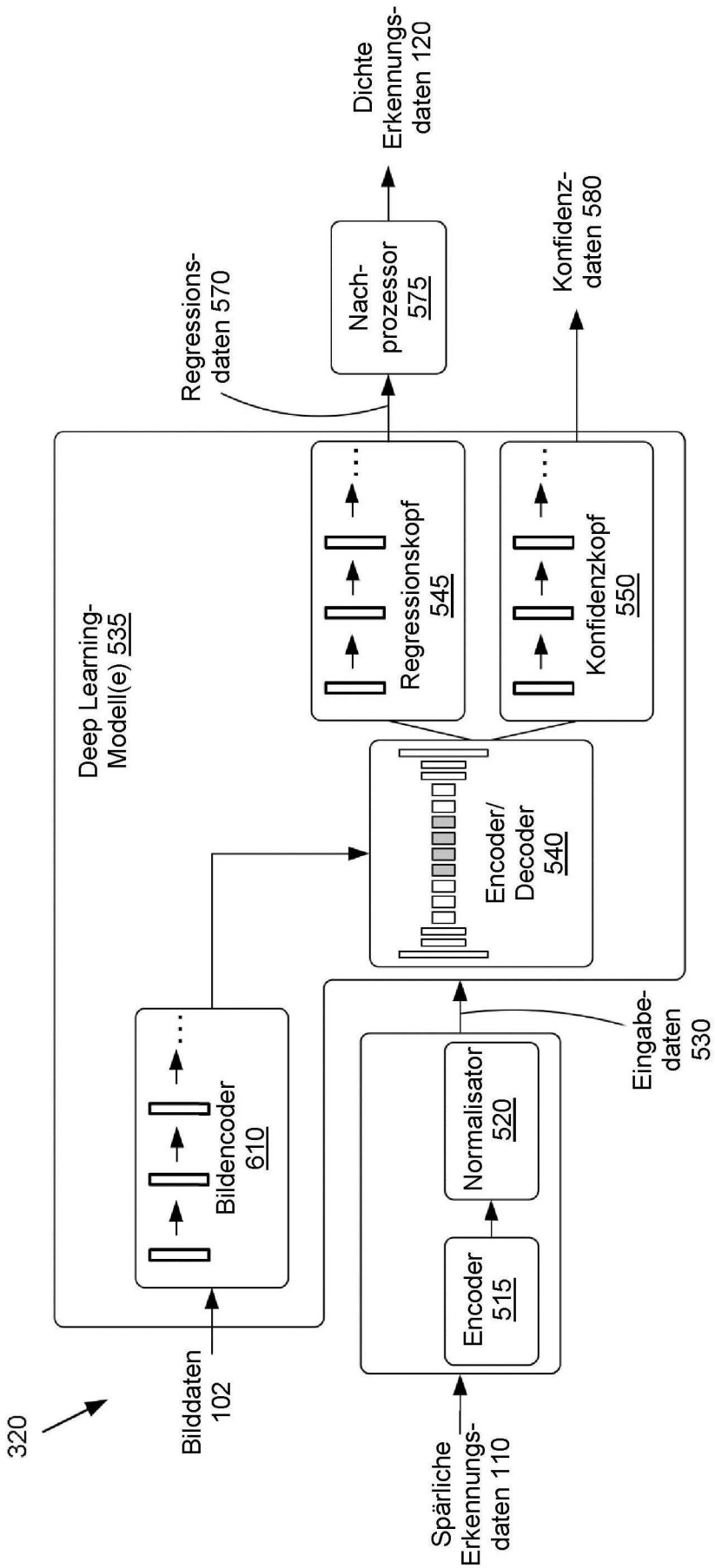

320
Bilddaten 102
Spärliche Erkennungs-daten 110
Deep Learning-Modell(e) 535
Bildencoder 610
Encoder 515
Normalisator 520
Eingabe-daten 530
Encoder/ Decoder 540
Regressionskopf 545
Konfidenzkopf 550
Regressions-daten 570
Konfidenz-daten 580
Nach-prozessor 575
Dichte Erkennungs-daten 120

Fig. 6

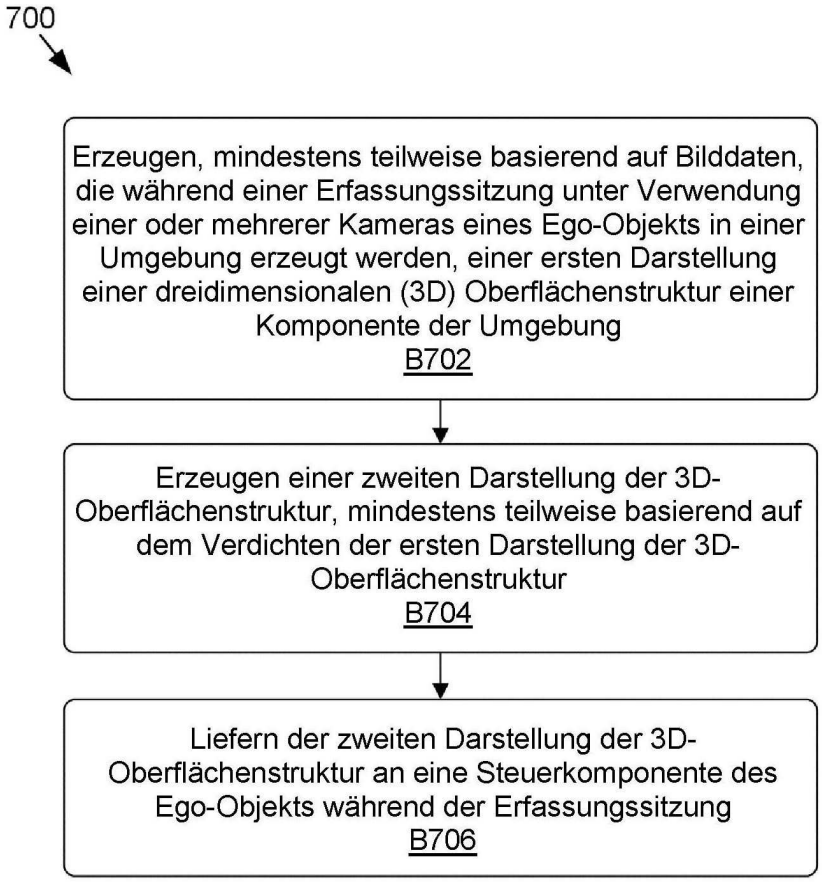

700
Erzeugen, mindestens teilweise basierend auf Bilddaten, die während einer Erfassungssitzung unter Verwendung einer oder mehrerer Kameras eines Ego-Objekts in einer Umgebung erzeugt werden, einer ersten Darstellung einer dreidimensionalen (3D) Oberflächenstruktur einer Komponente der Umgebung
B702
Erzeugen einer zweiten Darstellung der 3D-Oberflächenstruktur, mindestens teilweise basierend auf dem Verdichten der ersten Darstellung der 3D-Oberflächenstruktur
B704
Liefern der zweiten Darstellung der 3D-Oberflächenstruktur an eine Steuerkomponente des Ego-Objekts während der Erfassungssitzung
B706

Fig. 7

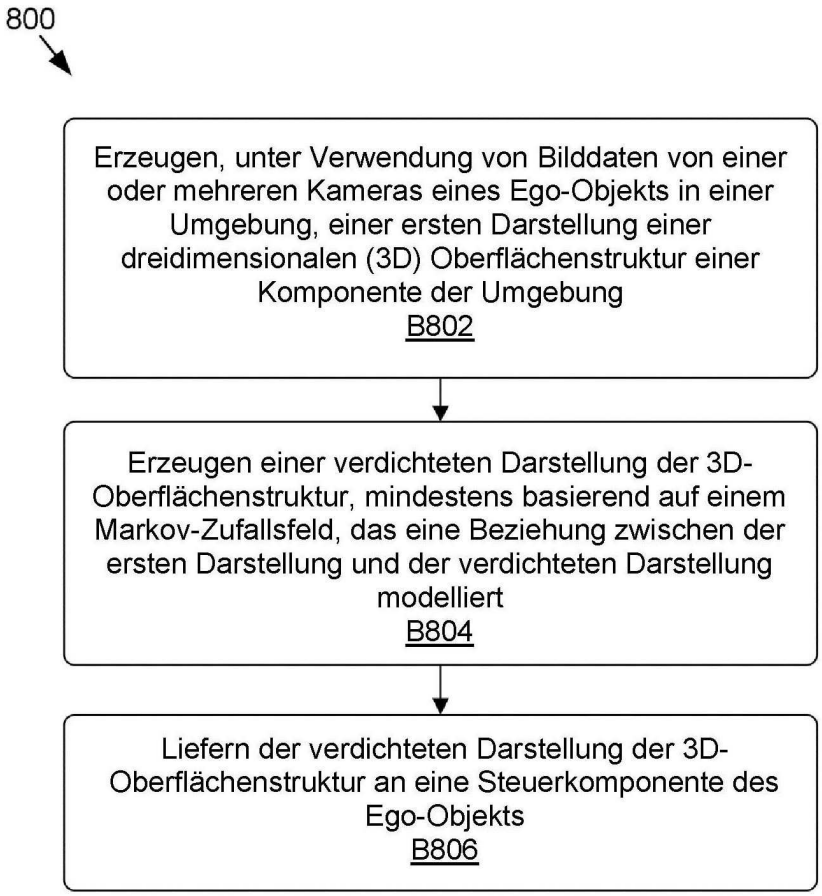

800
Erzeugen, unter Verwendung von Bilddaten von einer oder mehreren Kameras eines Ego-Objekts in einer Umgebung, einer ersten Darstellung einer dreidimensionalen (3D) Oberflächenstruktur einer Komponente der Umgebung
B802
Erzeugen einer verdichteten Darstellung der 3D-Oberflächenstruktur, mindestens basierend auf einem Markov-Zufallsfeld, das eine Beziehung zwischen der ersten Darstellung und der verdichteten Darstellung modelliert
B804
Liefern der verdichteten Darstellung der 3D-Oberflächenstruktur an eine Steuerkomponente des Ego-Objekts
B806

Fig. 8

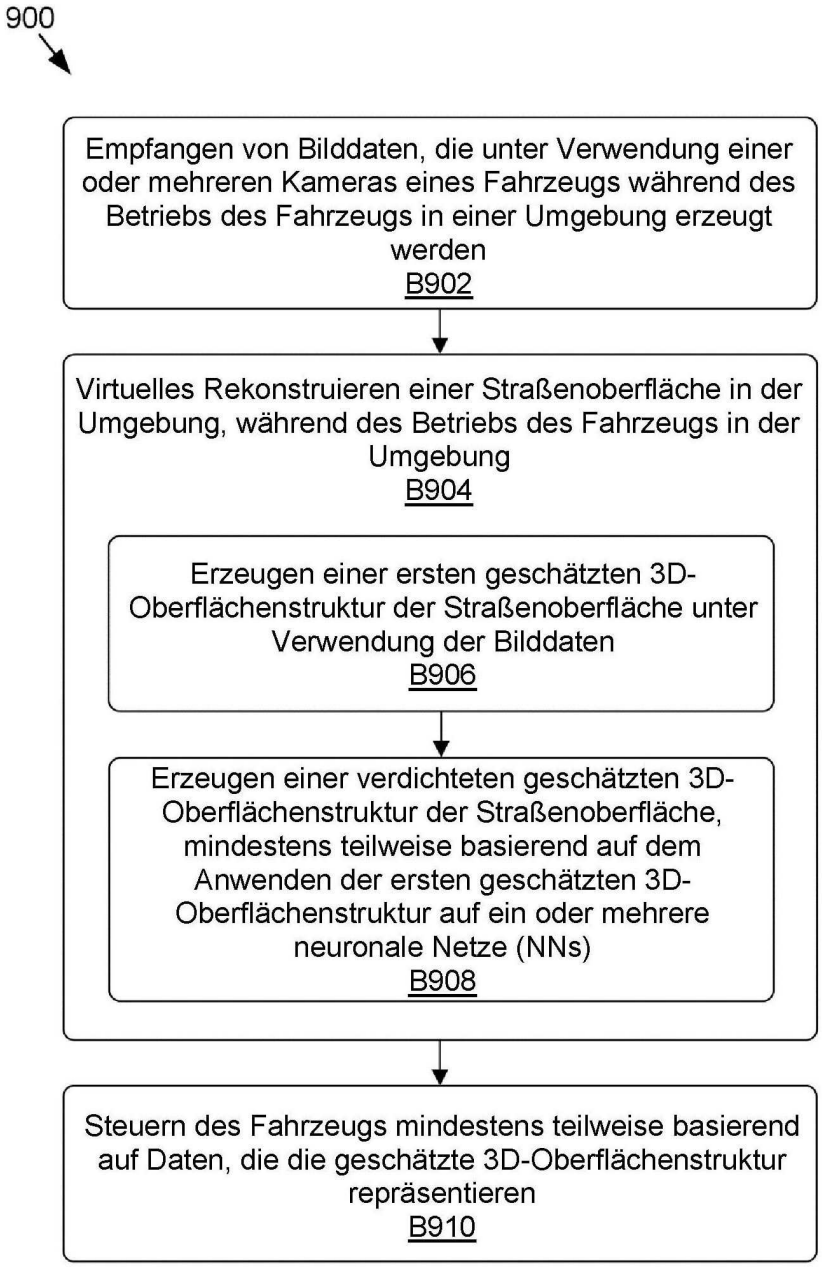

900
Empfangen von Bilddaten, die unter Verwendung einer oder mehreren Kameras eines Fahrzeugs während des Betriebs des Fahrzeugs in einer Umgebung erzeugt werden
B902
Virtuelles Rekonstruieren einer Straßenoberfläche in der Umgebung, während des Betriebs des Fahrzeugs in der Umgebung
B904
Erzeugen einer ersten geschätzten 3D-Oberflächenstruktur der Straßenoberfläche unter Verwendung der Bilddaten
B906
Erzeugen einer verdichteten geschätzten 3D-Oberflächenstruktur der Straßenoberfläche, mindestens teilweise basierend auf dem Anwenden der ersten geschätzten 3D-Oberflächenstruktur auf ein oder mehrere neuronale Netze (NNs)
B908
Steuern des Fahrzeugs mindestens teilweise basierend auf Daten, die die geschätzte 3D-Oberflächenstruktur repräsentieren
B910

Fig. 9

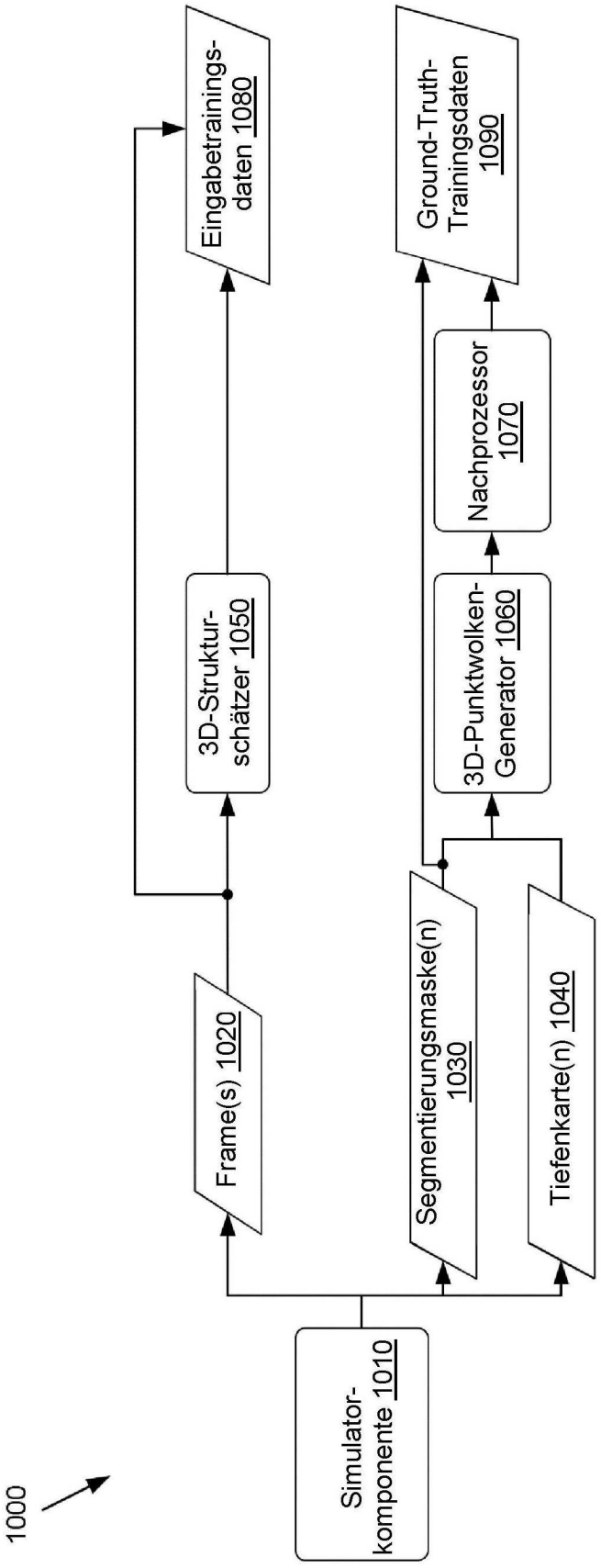

1000
Simulator-komponente 1010
Frame(s) 1020
Segmentierungsmaske(n) 1030
Tiefenkarte(n) 1040
3D-Struktur-schätzer 1050
3D-Punktwolken-Generator 1060
Nachprozessor 1070
Eingabetrainings-daten 1080
Ground-Truth-Trainingsdaten 1090

Fig. 10

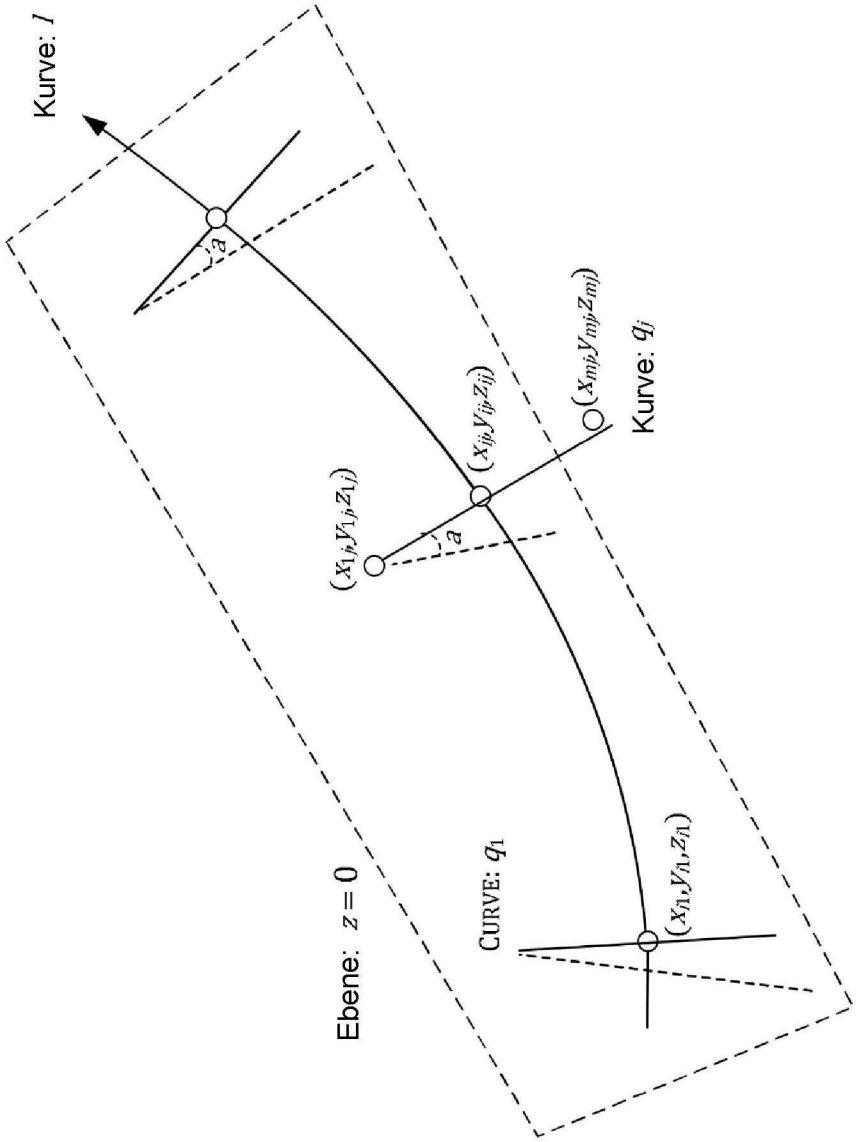

Kurve: l
a
(x_{mj}, y_{mj}, z_{mj})
Kurve: q_j
(x_{jj}, y_{jj}, z_{jj})
(x_{1j}, y_{1j}, z_{1j})
a
Ebene: $z = 0$
CURVE: q_1
(x_{11}, y_{11}, z_{11})

Fig. 11

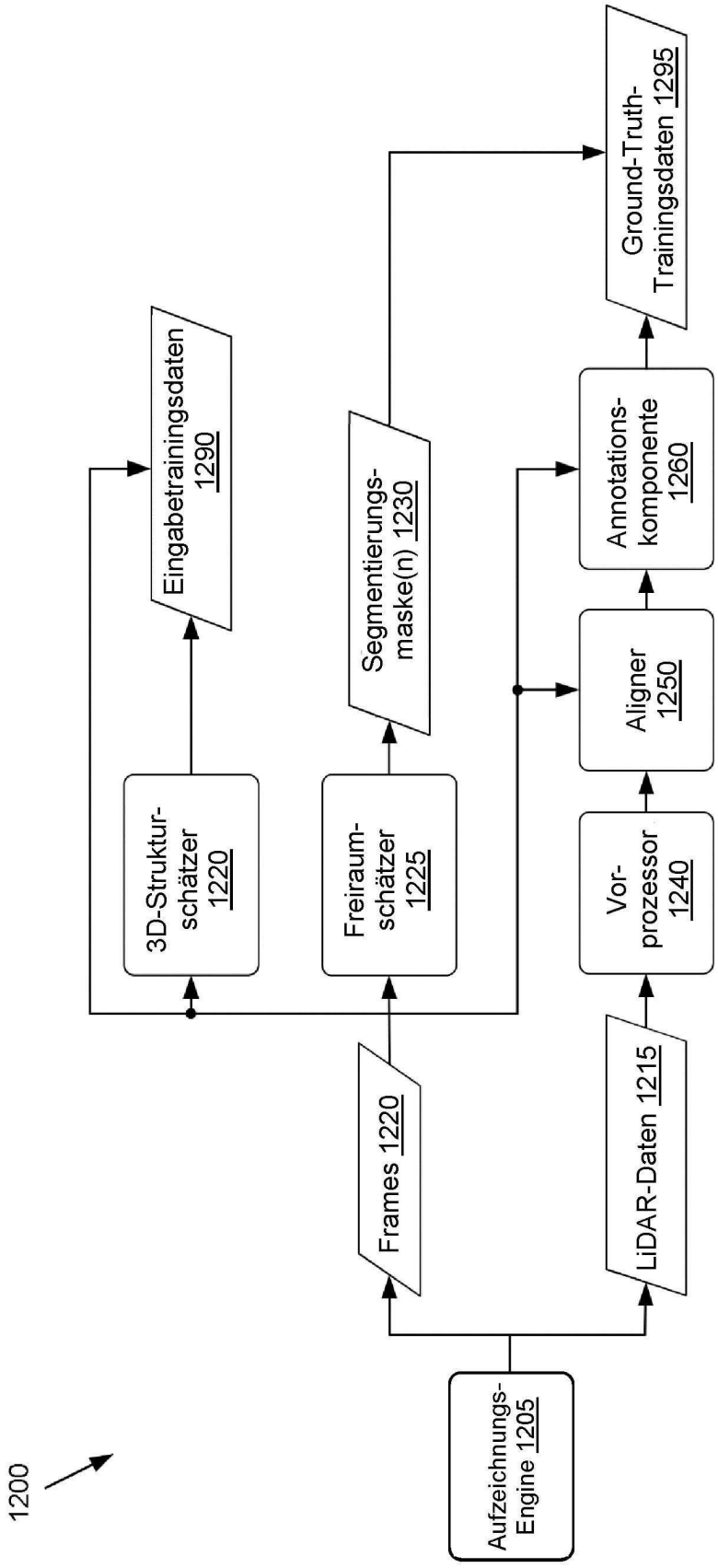

1200
Aufzeichnungs-Engine 1205
Frames 1220
LiDAR-Daten 1215
3D-Struktur-schätzer 1220
Freiraum-schätzer 1225
Vor-prozessor 1240
Eingabetrainingsdaten 1290
Segmentierungs-maske(n) 1230
Aligner 1250
Annotations-komponente 1260
Ground-Truth-Trainingsdaten 1295

Fig. 12

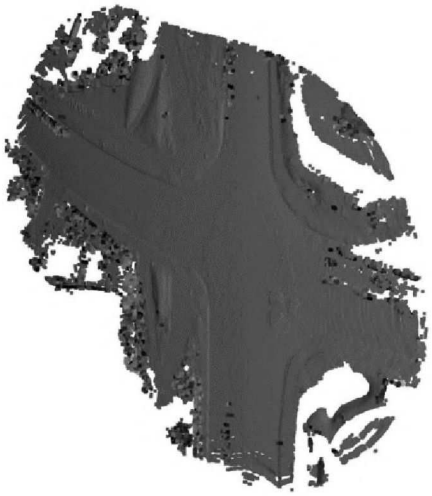


Fig. 13B

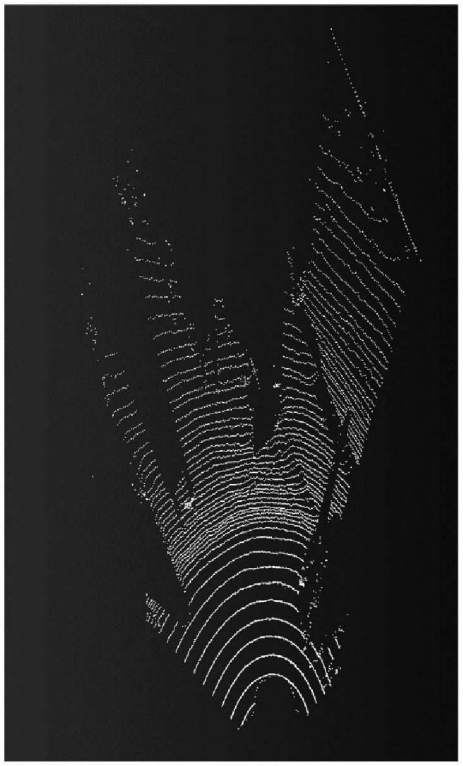


Fig. 13A

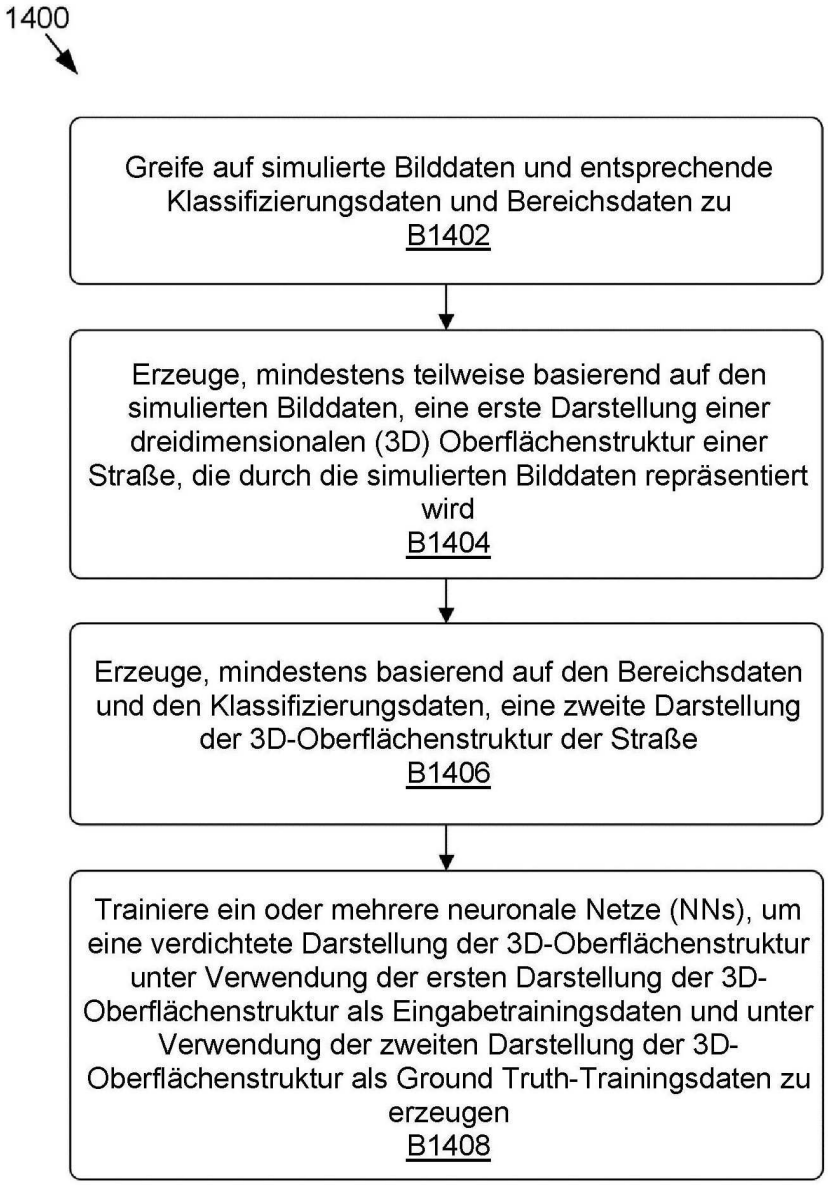

1400
Greife auf simulierte Bilddaten und entsprechende Klassifizierungsdaten und Bereichsdaten zu
B1402
Erzeuge, mindestens teilweise basierend auf den simulierten Bilddaten, eine erste Darstellung einer dreidimensionalen (3D) Oberflächenstruktur einer Straße, die durch die simulierten Bilddaten repräsentiert wird
B1404
Erzeuge, mindestens basierend auf den Bereichsdaten und den Klassifizierungsdaten, eine zweite Darstellung der 3D-Oberflächenstruktur der Straße
B1406
Trainiere ein oder mehrere neuronale Netze (NNs), um eine verdichtete Darstellung der 3D-Oberflächenstruktur unter Verwendung der ersten Darstellung der 3D-Oberflächenstruktur als Eingabetrainingsdaten und unter Verwendung der zweiten Darstellung der 3D-Oberflächenstruktur als Ground Truth-Trainingsdaten zu erzeugen
B1408

Fig. 14

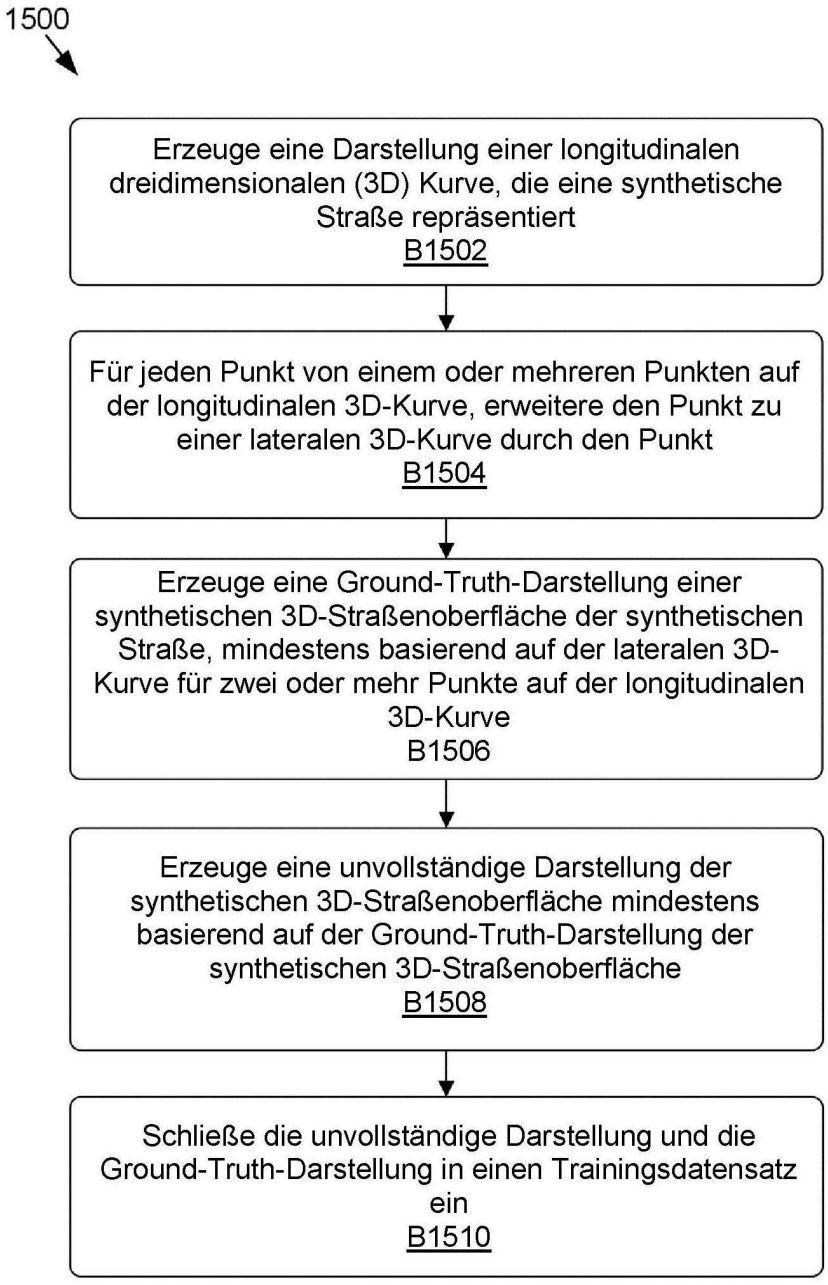

1500
Erzeuge eine Darstellung einer longitudinalen dreidimensionalen (3D) Kurve, die eine synthetische Straße repräsentiert
B1502
Für jeden Punkt von einem oder mehreren Punkten auf der longitudinalen 3D-Kurve, erweitere den Punkt zu einer lateralen 3D-Kurve durch den Punkt
B1504
Erzeuge eine Ground-Truth-Darstellung einer synthetischen 3D-Straßenoberfläche der synthetischen Straße, mindestens basierend auf der lateralen 3D-Kurve für zwei oder mehr Punkte auf der longitudinalen 3D-Kurve
B1506
Erzeuge eine unvollständige Darstellung der synthetischen 3D-Straßenoberfläche mindestens basierend auf der Ground-Truth-Darstellung der synthetischen 3D-Straßenoberfläche
B1508
Schließe die unvollständige Darstellung und die Ground-Truth-Darstellung in einen Trainingsdatensatz ein
B1510

Fig. 15

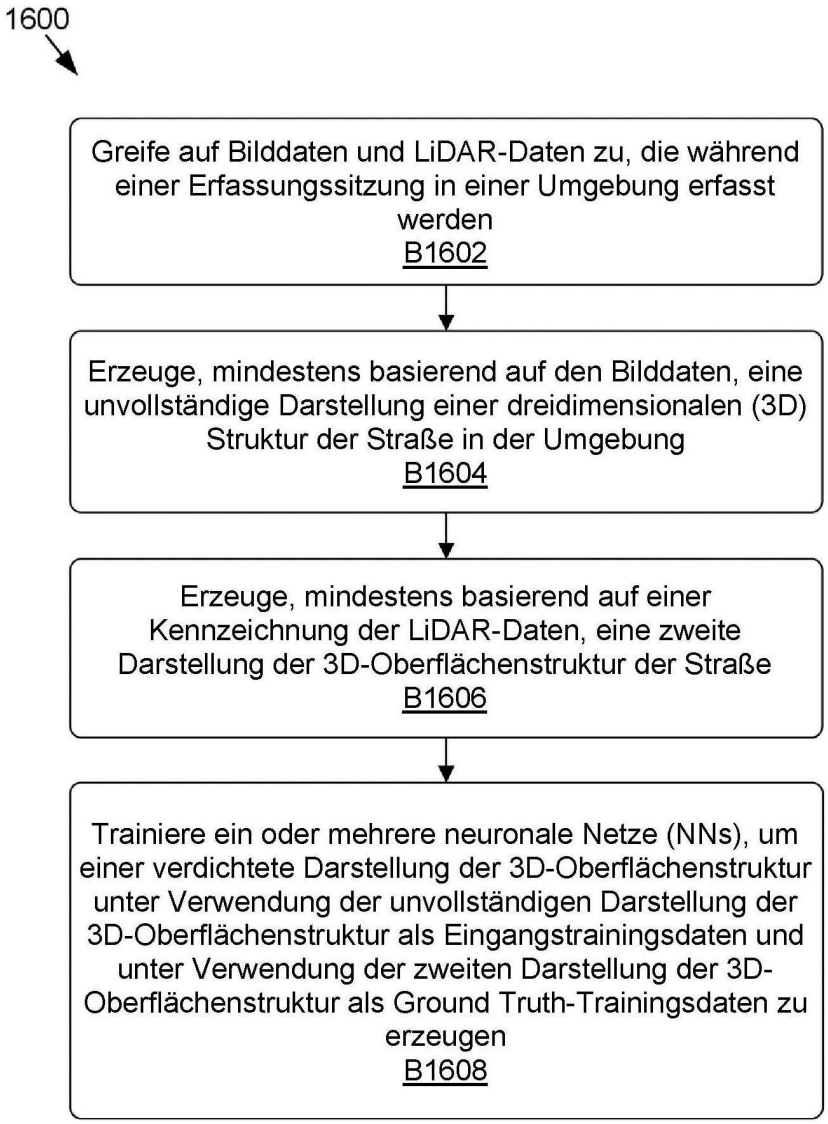

1600
Greife auf Bilddaten und LiDAR-Daten zu, die während einer Erfassungssitzung in einer Umgebung erfasst werden
B1602
Erzeuge, mindestens basierend auf den Bilddaten, eine unvollständige Darstellung einer dreidimensionalen (3D) Struktur der Straße in der Umgebung
B1604
Erzeuge, mindestens basierend auf einer Kennzeichnung der LiDAR-Daten, eine zweite Darstellung der 3D-Oberflächenstruktur der Straße
B1606
Trainiere ein oder mehrere neuronale Netze (NNs), um einer verdichtete Darstellung der 3D-Oberflächenstruktur unter Verwendung der unvollständigen Darstellung der 3D-Oberflächenstruktur als Eingangstrainingsdaten und unter Verwendung der zweiten Darstellung der 3D-Oberflächenstruktur als Ground Truth-Trainingsdaten zu erzeugen
B1608

Fig. 16

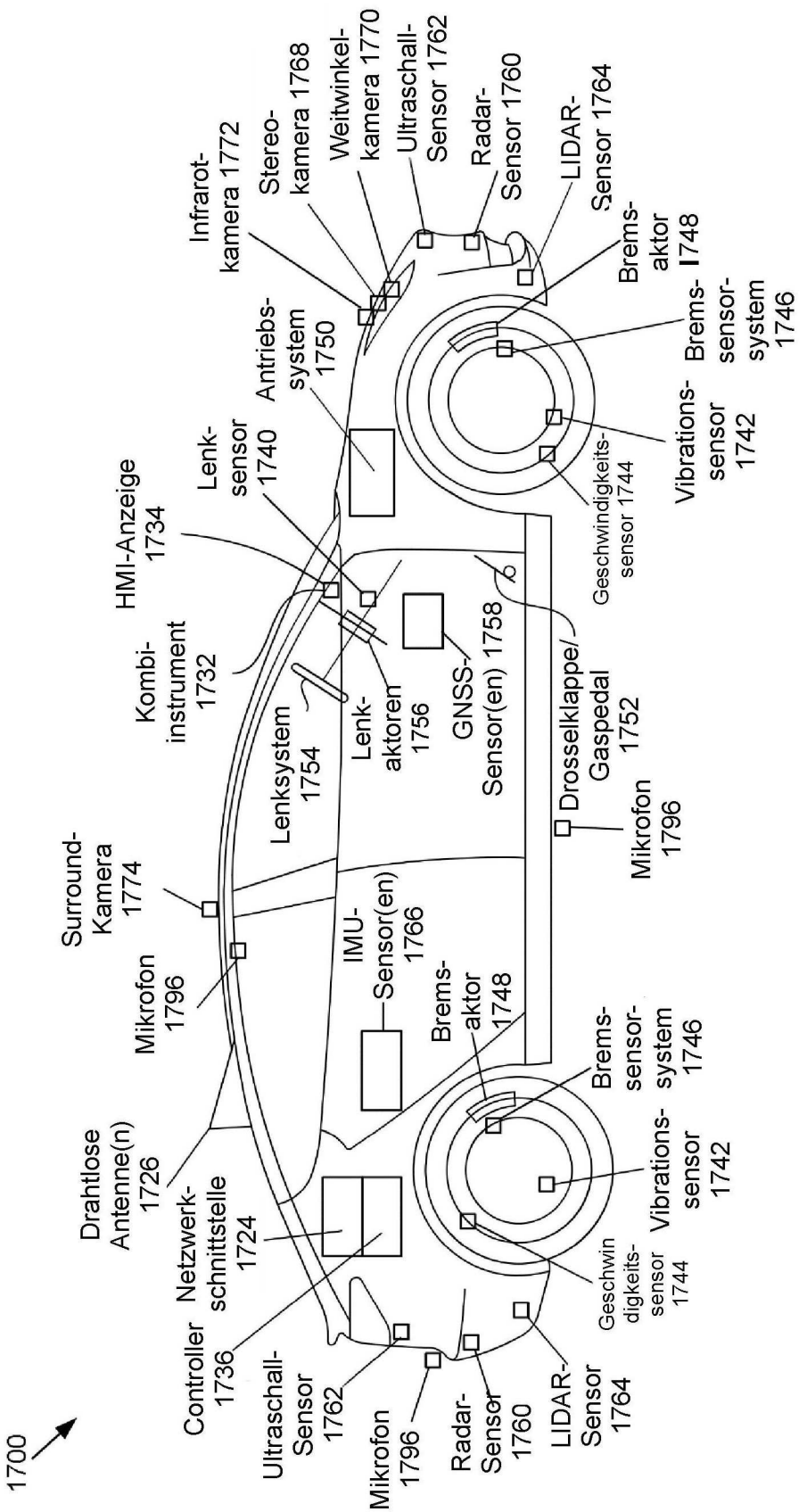

1700
Controller 1736
Netzwerk-schnittstelle 1724
Drahtlose Antenne(n) 1726
Surround-Kamera 1774
Mikrofon 1796
Kombi-instrument 1732
HMI-Anzeige 1734
Lenk-sensor 1740
Lenksystem 1754
Lenk-aktoren 1756
GNSS-Sensor(en) 1758
IMU-Sensor(en) 1766
Antriebs-system 1750
Infrarot-kamera 1772
Stereo-kamera 1768
Weitwinkel-kamera 1770
Ultraschall-Sensor 1762
Radar-Sensor 1760
LIDAR-Sensor 1764
Brems-aktor 1748
Brems-sensor-system 1746
Vibrations-sensor 1742
Geschwindigkeits-sensor 1744
Drosselklappe/ Gaspedal 1752
Mikrofon 1796
Brems-aktor 1748
Brems-sensor-system 1746
Vibrations-sensor 1742
Geschwin digkeits-sensor 1744
Ultraschall-Sensor 1762
Mikrofon 1796
Radar-Sensor 1760
LIDAR-Sensor 1764

Fig. 17A

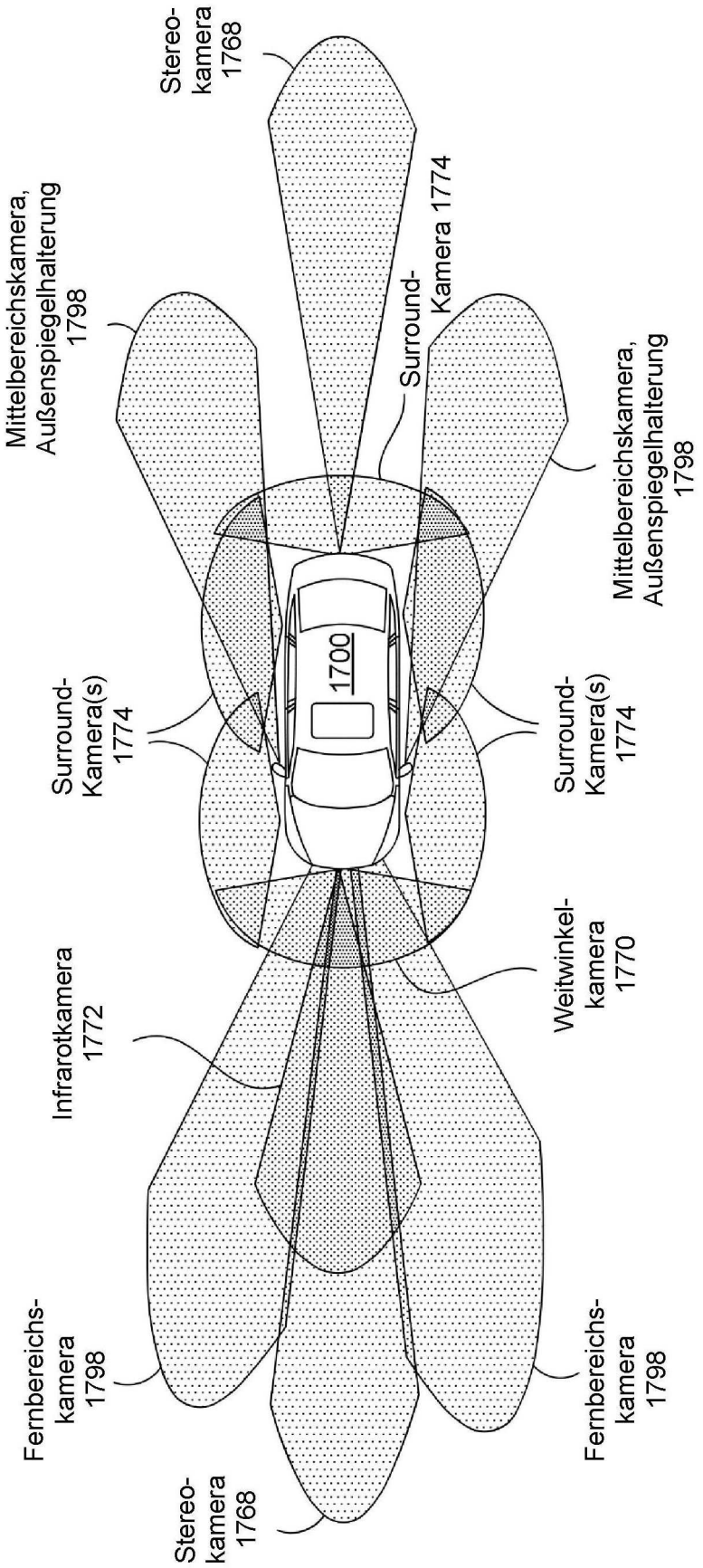

Stereo-kamera 1768
Mittelbereichskamera, Außenspiegelhalterung 1798
Surround-Kamera 1774
Mittelbereichskamera, Außenspiegelhalterung 1798
1700
Surround-Kamera(s) 1774
Surround-Kamera(s) 1774
Infrarotkamera 1772
Weitwinkel-kamera 1770
Fernbereichs-kamera 1798
Fernbereichs-kamera 1798
Stereo-kamera 1768

Fig. 17B

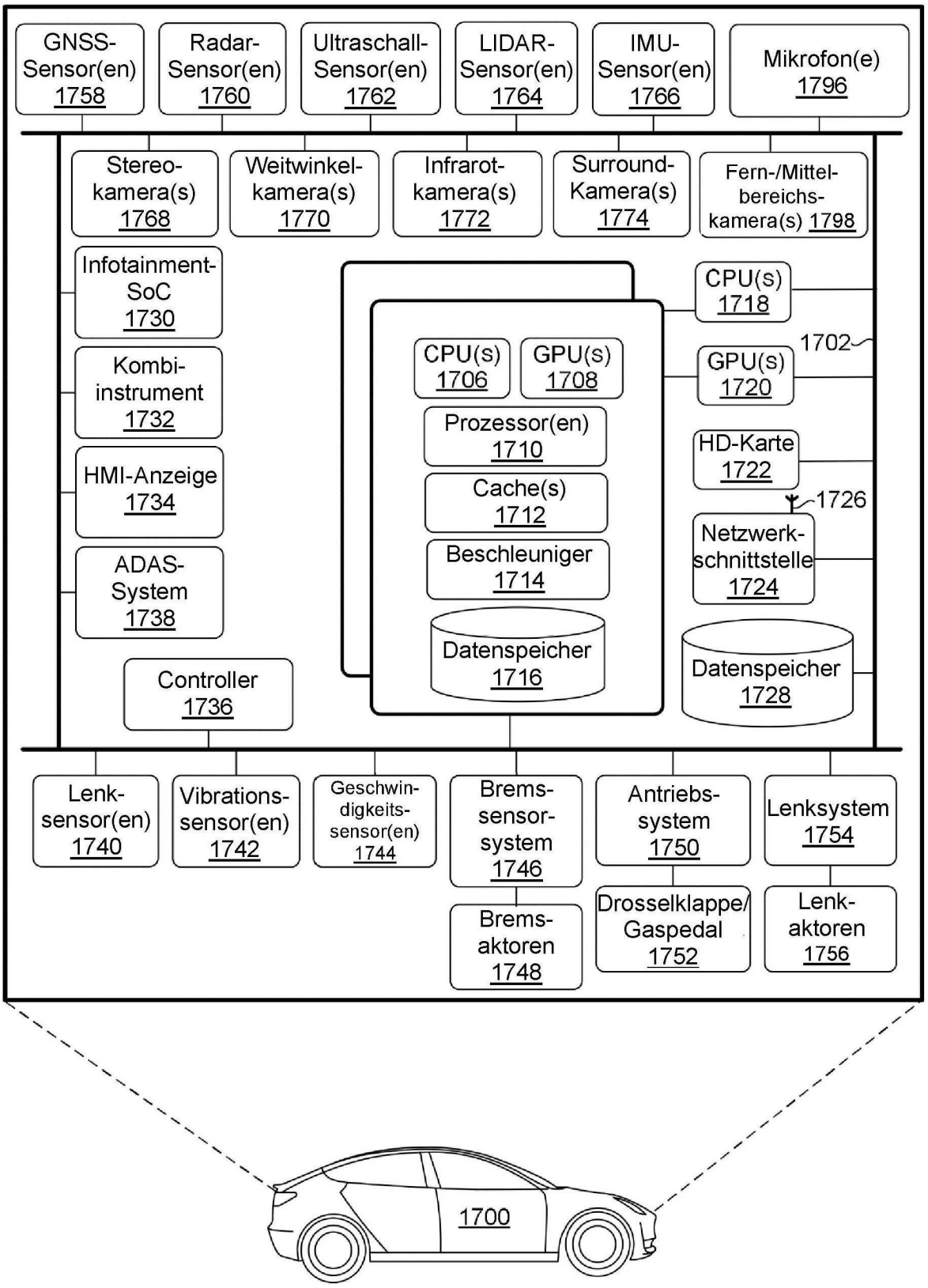

GNSS-Sensor(en) 1758
Radar-Sensor(en) 1760
Ultraschall-Sensor(en) 1762
LIDAR-Sensor(en) 1764
IMU-Sensor(en) 1766
Mikrofon(e) 1796
Stereo-kamera(s) 1768
Weitwinkel-kamera(s) 1770
Infrarot-kamera(s) 1772
Surround-Kamera(s) 1774
Fern-/Mittel-bereichs-kamera(s) 1798
Infotainment-SoC 1730
Kombi-instrument 1732
HMI-Anzeige 1734
ADAS-System 1738
Controller 1736
CPU(s) 1706
GPU(s) 1708
Prozessor(en) 1710
Cache(s) 1712
Beschleuniger 1714
Datenspeicher 1716
CPU(s) 1718
GPU(s) 1720
1702
HD-Karte 1722
1726
Netzwerk-schnittstelle 1724
Datenspeicher 1728
Lenk-sensor(en) 1740
Vibrations-sensor(en) 1742
Geschwin-digkeits-sensor(en) 1744
Brems-sensor-system 1746
Brems-aktoren 1748
Antriebs-system 1750
Drosselklappe/ Gaspedal 1752
Lenksystem 1754
Lenk-aktoren 1756
1700

Fig. 17C

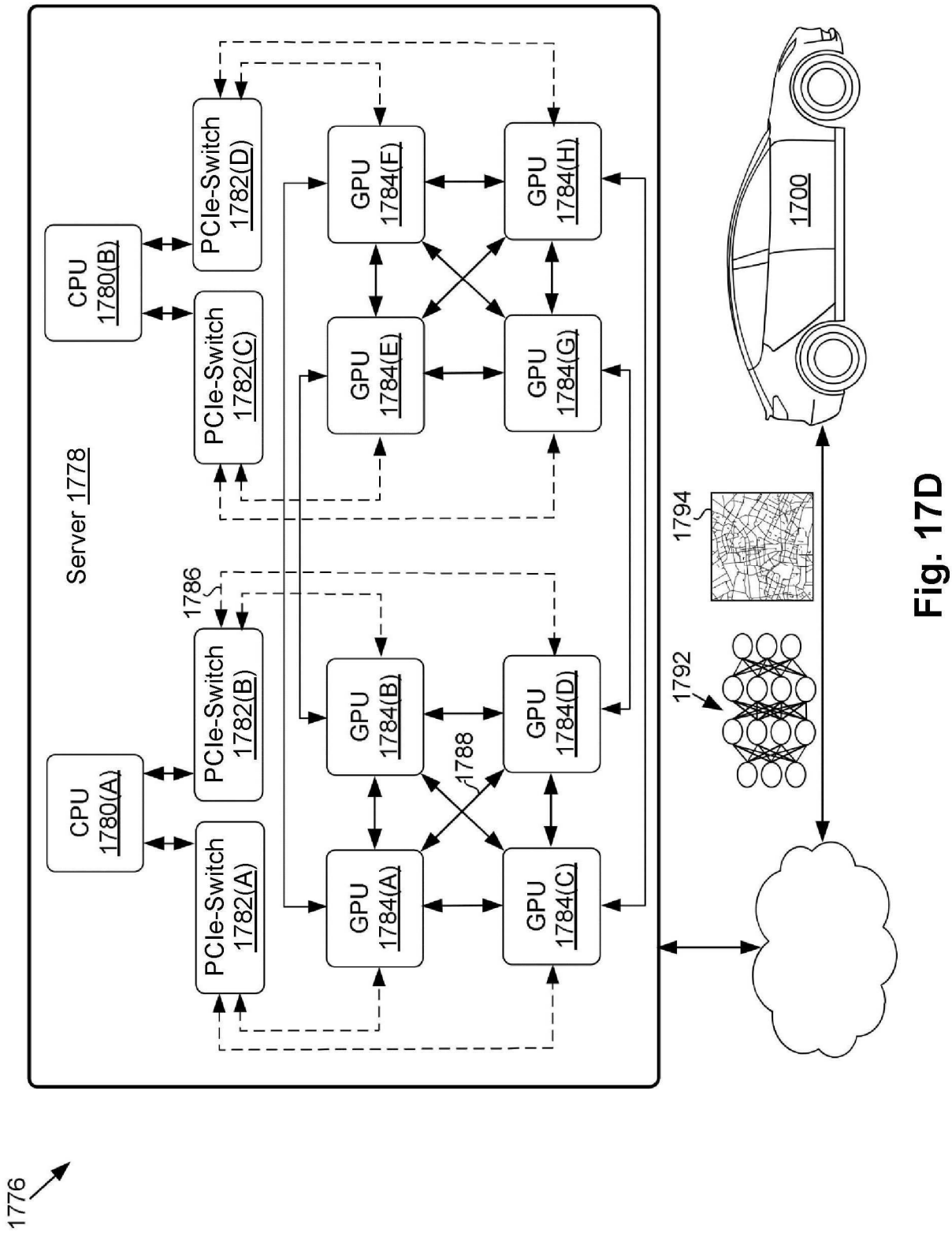

1776
Server 1778
CPU 1780(A)
CPU 1780(B)
PCIe-Switch 1782(A)
PCIe-Switch 1782(B)
PCIe-Switch 1782(C)
PCIe-Switch 1782(D)
1786
GPU 1784(A)
GPU 1784(B)
GPU 1784(C)
GPU 1784(D)
GPU 1784(E)
GPU 1784(F)
GPU 1784(G)
GPU 1784(H)
1788
1700
1794
1792

Fig. 17D

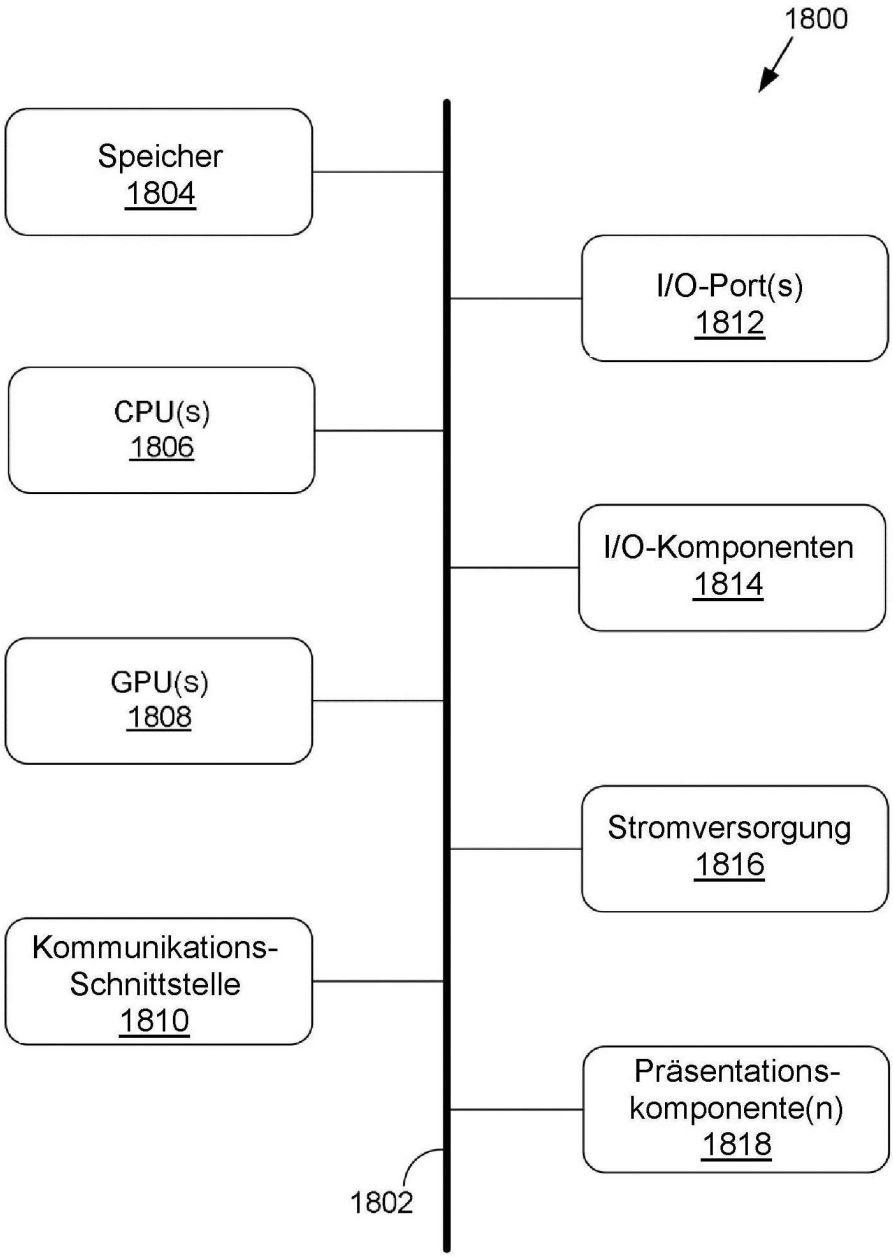

1800
Speicher
1804
I/O-Port(s)
1812
CPU(s)
1806
I/O-Komponenten
1814
GPU(s)
1808
Stromversorgung
1816
Kommunikations-
Schnittstelle
1810
Präsentations-
komponente(n)
1818
1802

Fig. 18

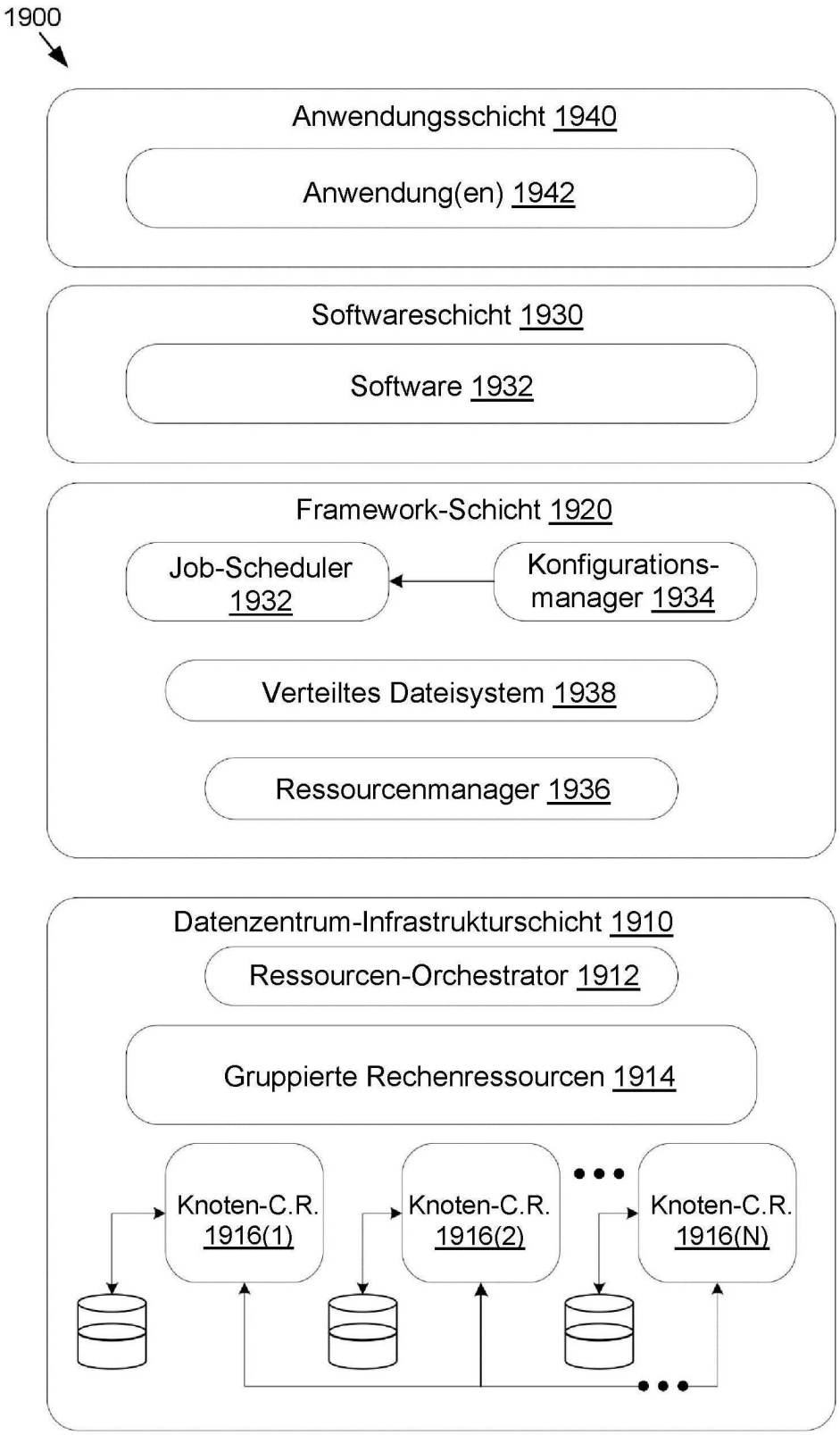


Fig. 19

 Das Dokument wurde durch die Firma Luminess hergestellt.