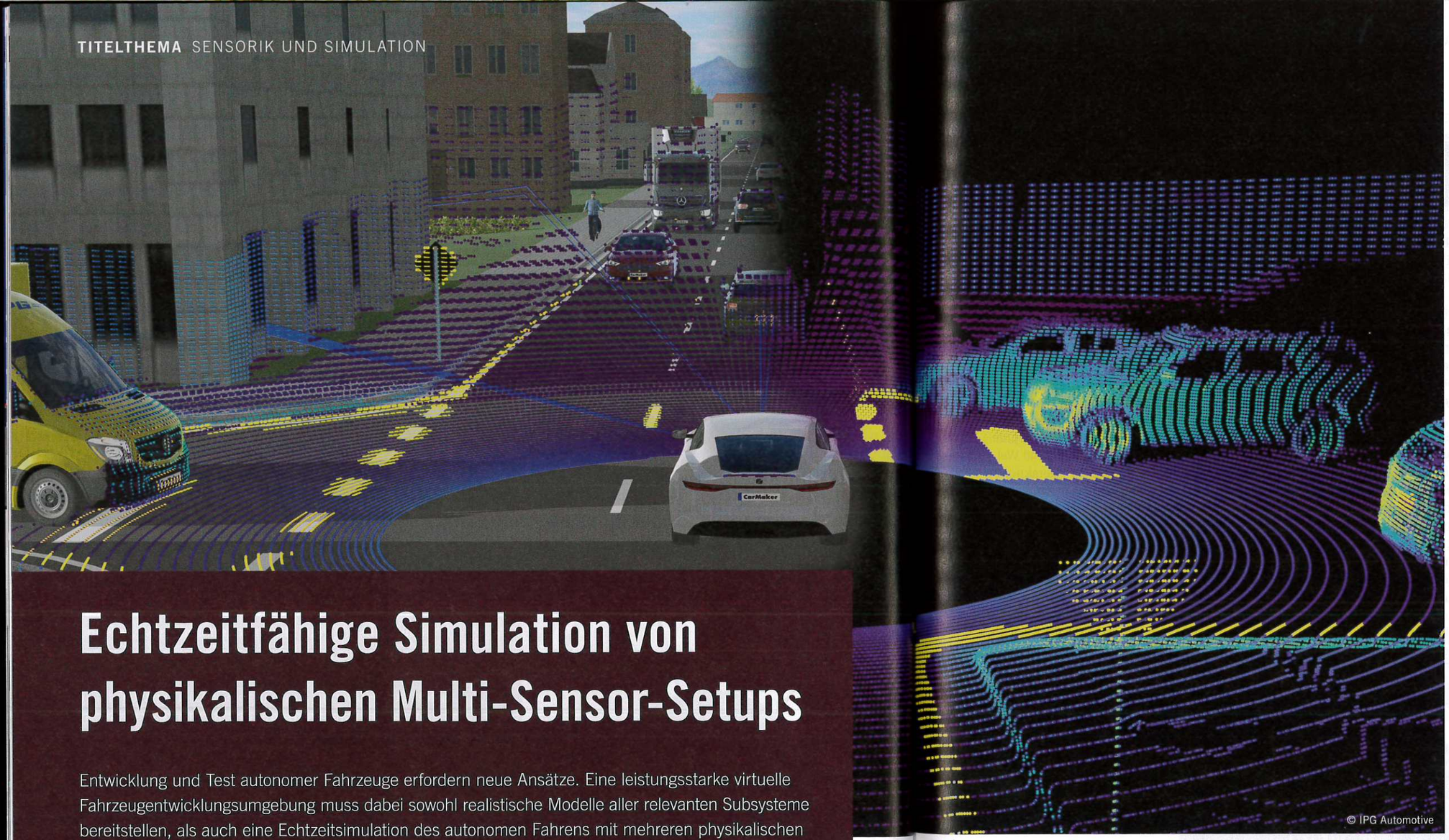


© IPG Automotive

Abbild eines realen Fahrzeugs mit realitätsgetreuem Verhalten und verfügt über entsprechende Modelle aller relevanten realen Komponenten. Bei Hardware-in-the-Loop(HiL)-Tests werden reale physische Hardwarekomponenten in einen virtuellen Fahrzeugprototyp integriert, weshalb die Echtzeitfähigkeit einer virtuellen Fahrzeugentwicklungsumgebung unverzichtbar ist. Der virtuelle Fahrzeugprototyp bewegt sich innerhalb einer simulierten 3-D-Umgebung, die es ermöglicht, die Interaktion des physischen Fahrzeugs mit der realen Welt realistisch abzubilden. Virtuelle Sensoren bilden die Perzeption der 3-D-Umgebung ab und liefern Input für weitere Algorithmen der Objekterkennung, Sensorfusion, Manöverplanung und Regelung. Eine flexible und leistungsstarke virtuelle Fahrzeugentwicklungsumgebung muss daher ein Multi-Sensor-Setup eines typischen autonomen Fahrzeugs effizient unterstützen. Die performante Simulation realistischer Multi-Sensor-Setups verlangt häufig den Einsatz mehrerer Grafikprozessoren (Graphics Processing Units, GPUs), um eine beschleunigte Berechnung der Sensorsimulation zu ermöglichen und den Anforderungen für HiL-Tests in Echtzeit gerecht zu werden.

Echtzeitfähige Simulation von physikalischen Multi-Sensor-Setups

Entwicklung und Test autonomer Fahrzeuge erfordern neue Ansätze. Eine leistungsstarke virtuelle Fahrzeugentwicklungsumgebung muss dabei sowohl realistische Modelle aller relevanten Subsysteme bereitstellen, als auch eine Echtzeitsimulation des autonomen Fahrens mit mehreren physikalischen Sensormodellen ermöglichen. IPG Automotive beschreibt, wie diese Recheneffizienz durch den Einsatz von Multi-GPU-Parallelisierung erreicht werden kann.

AUTOREN

Dr. Natalya Ahn
ist Business Development Manager ADAS und Autonomes Fahren bei der IPG Automotive GmbH in Karlsruhe.

Dr. Andreas Höfer
ist Produktmanager Simulation Software bei der IPG Automotive GmbH in Karlsruhe.

Martin Herrmann
ist Business Development Manager ADAS und Autonomes Fahren bei der IPG Automotive GmbH in Frankfurt.

Dr. Christian Donn
ist Manager Business Development bei der IPG Automotive GmbH in Karlsruhe.

ENTWICKLUNGSUMGEBUNG FÜR AUTONOME FAHRZEUGE

Autonomes Fahren stellt vor allem Fahrzeugentwickler vor viele neue Herausforderungen. Autonome Fahrzeuge der SAE-Level 4 oder 5 sind üblicherweise mit 20 bis 30 verschiedenen Sensoren ausgestattet, von Kameras, Lidar und Radar bis hin zu Ultraschall [1]. Eine so große Zusammensetzung verschiedener Sensoren ist nötig, um einen verlässlichen 360°-Blick sowie Redundanz herzustellen und so die fehlerfreie Funktion eines autonomen Fahrzeugs zu gewährleisten. In der Industrie herrscht ein allgemeines Verständnis, dass es nicht möglich ist, autonome Fahrzeuge mit so vielen Sensoren und komplexen Software-Stacks nur mithilfe von konventionellen realen Fahrversuchen zu entwickeln und zu testen. Für jede Million auf öffentlichen Straßen in der Realität gefahrener Meilen hat Waymo eine Milliarde simulierter Meilen zurückgelegt [2]. Eine umfassende und skalierbare Simulationsplattform, die virtuelle Fahrzeugentwicklung ermöglicht, ist von größter Wichtigkeit.

Die virtuelle Fahrzeugentwicklung basiert auf einem virtuellen Fahrzeugprototyp. Dieser ist ein vollständiges

SENSORSIMULATION IM AUTOMOBILBEREICH

Perzeptionssensoren für Fahrzeuge, die für die Erkennung der Umgebung notwendig sind, können entweder aktiv oder passiv sein. Aktive Perzeptionssensoren senden aktiv Wellen aus und nutzen empfangene Wellen, um Informationen zur Umgebung zu erlangen. Typische Beispiele aktiver Perzeptionssensoren sind Ultraschall-, Radar- und Lidarsensoren. Passive Perzeptionssensoren geben für die Perzeption nicht aktiv Energie in die Umgebung ab und sind auf externe Strahlenquellen angewiesen. Ein typisches Beispiel hierfür sind Kameras.

PURPOSE-DRIVEN FIDELITY FÜR DIE SENSORMODELLIERUNG

Abhängig von der Entwicklungsphase oder dem Anwendungsfall können bestimmte Modelle von Fahrzeugkomponenten hochdetailliert sein, während andere einfacher abgebildet werden können, um die größtmögliche Perfor-

BILD 1 Ansatz der Purpose-Driven Fidelity für die Sensormodellierung (© IPG Automotive)

manz zu ermöglichen. Dem Ansatz der Purpose-Driven Fidelity folgend kann die Abbildung von Perzeptionssensoren mithilfe von drei verschiedenen Sensormodellklassen umgesetzt werden: ideal, phänomenologisch und physikalisch [3], **BILD 1**.

Ideale Sensormodelle geben eine Liste von relevanten detektierten Objekten aus. Die Informationen, die dafür benötigt werden, werden unmittelbar aus dem Simulationsmodell (Ground Truth) extrahiert. So kann eine technologieunabhängige ideale Umgebungserfassung erfolgen. Dabei wird weder die Sensorphysik betrachtet, noch werden sensortypische Fehler in der Objektliste abgebildet.

Auch die phänomenologischen Sensormodelle liefern eine Objektliste. In diesem Fall werden die Informationen durch physikalische Effekte und/oder technologiespezifische Fehler angereichert, anstatt eine reine Ground-Truth-Extraktion vorzunehmen. Auf diese Weise ergibt sich für die gewählte Sensortechnologie eine realitätsnahe Objektliste.

Die Aufgabe von physikalischen Sensormodellen ist es, Eingangsdaten für die Perzeptionsalgorithmen des Sensors bereitzustellen. Dies können zum Beispiel Bilddaten für die Kamerasimulation oder Lidarpunktwolken sein. Für die Generierung solcher Rohsignale müssen detaillierte physikalische Effekte sowie die Geometrie und die Materialeigenschaften der Objekte berücksichtigt werden. Physikalische Sensormodelle sind die realistischsten und naturgemäß rechenintensivsten Sensormodelle. In der offenen Integrations- und Testplattform CarMaker werden physikalische Sensormodelle Raw Signal Interfaces (RSIs) genannt. CarMaker nutzt die erhöhte Leistung von GPUs für eine effiziente Echtzeitsimulation der RSI-Sensormodelle, die ein Fokus dieses Artikels ist.

PHYSIKALISCHE MODELLIERUNG AKTIVER SENSOREN

Physikalische Sensormodelle aktiver Sensoren basieren häufig auf einem Raytracing-Ansatz. Dieser wird in verschiedenen Bereichen angewandt, darunter Visualisierung, Akustik und kabellose Kommunikation, in denen eine Ausbreitung von Wellen in einer 3-D-Umgebung simuliert werden muss [4]. Das Ziel von Raytracing ist die Berechnung der Wellenausbreitung mithilfe einzelner Strahlen und deren Reflexionen, um die Energiemenge zu berechnen, die von einem Empfänger empfangen wird, nachdem sie von einem Sender ausgesandt wurde. Je nach Materialeigenschaften und Wellenfrequenz kann sich die Interaktion mit der Umgebung dämpfend auf die ausgebreiteten Wellen auswirken.

Die in CarMaker simulierten RSIs nutzen den Raytracing-Ansatz für die Abbildung aktiver Sensoren: Das Ultraschall-RSI simuliert die Ausbreitung mechanischer Schalldruckwellen im virtuellen Szenario. Das Radar-RSI simuliert die Ausbreitung elektromagnetischer Wellen, und das Lidar-RSI simuliert die Ausbreitung von Licht. Für jeden empfangenen Strahl wird die Energiedämpfung aufgrund von Mehrwegeausbreitung berücksichtigt. Das Lidar-RSI ist aufgrund des Verhältnisses der Anzahl zurückkehrender Strahlen der verschiedenen Sensortechnologien üblicherweise sehr viel rechenintensiver als Radar- und Ultraschall-RSIs. Das Lidar-RSI erzeugt eine vergleichsweise große Menge von Sensordaten, die auf einer GPU berechnet werden.

Es ist möglich, verschiedene Arten einer bestimmen Sensortechnologie abzubilden, aber auch die Genauigkeit jedes einzelnen simulierten Sensors zu verändern, indem mehrere RSI-Parameter wie das Sichtfeld, die Reichweite oder die Anzahl der Strahlen variiert werden. Selbstverständlich hat die Parametrierung direkten Einfluss auf die Performanz, das heißt die benötigten Rechenressourcen.

PHYSIKALISCHE MODELLIERUNG EINER KAMERA

Obwohl der Raytracing-Ansatz sich auch für die Abbildung von Kamerasensoren eignet, ist dies aufgrund des enormen Rechenaufwands derzeit kaum in Echtzeit möglich, womit sich dieser Ansatz als ungeeignet für die virtuelle Fahrzeugentwicklung erweist. Daher wird oft der sehr viel schnellere Ansatz der Rasterung verfolgt, um für die physikalische Abbildung von Kameras 2-D-Bilder aus 3-D-Szenen zu rendern. Der Grundgedanke hinter 3-D-Rasterung ist die Rasterung von Dreiecken, um das Sichtbarkeitsproblem zu lösen [5].

Das Kamera-RSI in CarMaker kann verschiedene Linsentypen, Verzerrungen und Rauschen simulieren, mit denen ein typischer Kamerasensor in der Realität konfrontiert wird. Auch hier hat die Parametrierung eines Kamera-RSI direkten Einfluss auf die Performanz. Beispielsweise würde eine hochauflösende Kamera große Datenmengen generieren, die noch von der GPU heruntergeladen und weiterverarbeitet werden müssen. Zudem haben Fischaugenkameras eine erheblich höhere GPU-Nutzung im Vergleich zu gewöhnlichen Kameras. Der Grund hierfür ist die Notwendigkeit, mehrere Bilder mit verschiedenen Kameraausrichtungen zu rendern, die anschließend zu einem einzigen verzerrten Fischaugenbild zusammengefügt werden.

ANSÄTZE ZUR PARALLELISIERUNG

Aus der zuvor beschriebenen Komplexität folgt, dass die Simulation autonomer Fahrzeuge mit Multi-Sensor-Setups erheblich vom Einsatz von Multi-GPU-Parallelisierung profitieren kann. Diese kann auf verschiedene Arten erfolgen. Es ist sowohl möglich, auf einem Rechner, der über mehrere GPUs verfügt, parallele Berechnungen durchzuführen, als auch innerhalb eines Netzwerks mehrere Rechner gleichzeitig einzusetzen und die Rechenlast auf deren GPUs zu verteilen. Die höchstmögliche Performanz wird mit High-Performance-Computing-Clustern erzielt, die eine sehr hohe Anzahl von Prozessen

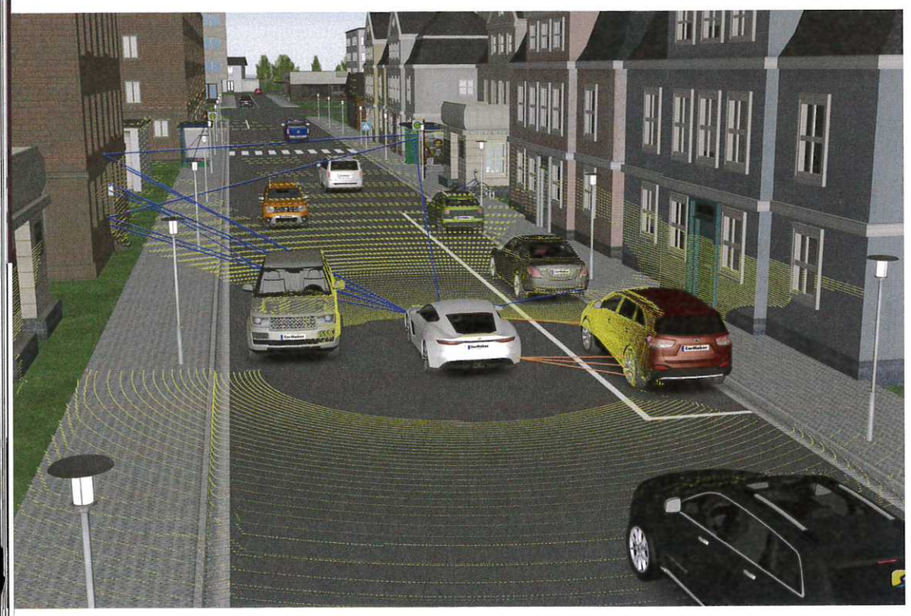
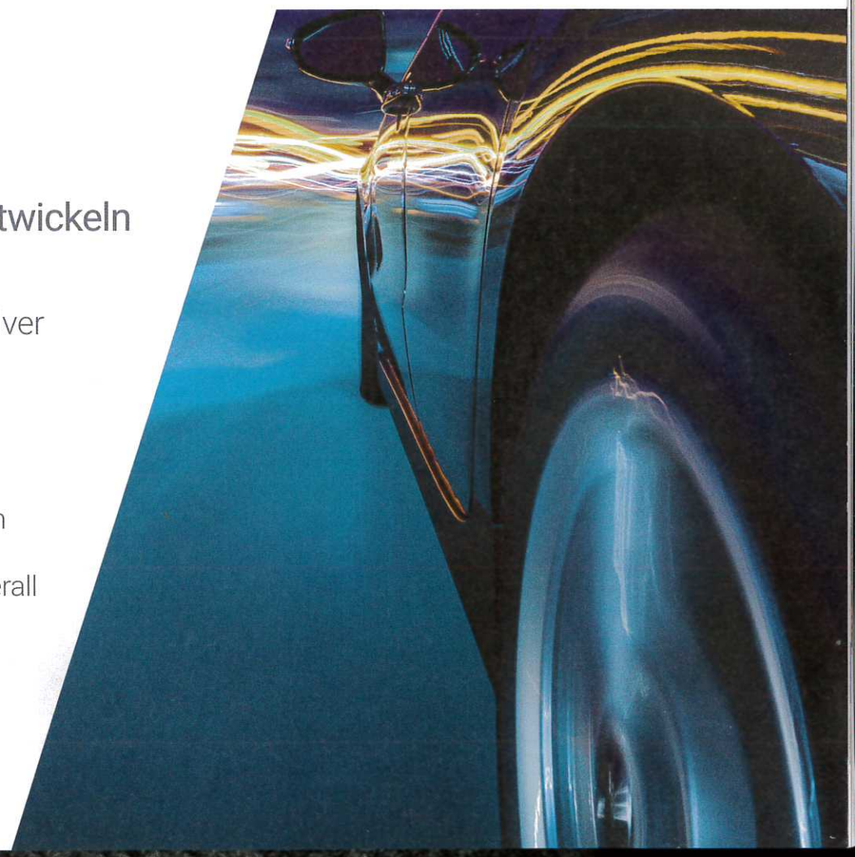


BILD 2 Simulation physikalischer Sensormodelle in CarMaker: empfangene Radar-RSI-Strahlen (blau), empfangene Lidar-RSI-Punktwolken (gelb), empfangene Ultraschall-RSI-Strahlen (orange) (© IPG Automotive)



parallel berechnen können. Dies ist mit CarMaker lokal und in der Cloud, etwa mit Microsoft Azure, möglich.

Für Echtzeitsimulationen spielt nicht nur die Anzahl der verfügbaren GPUs eine Rolle, sondern auch die Art der Anbindung der GPUs an den Zentralrechner. Automobilsensoren werden üblicherweise auf separaten GPUs simuliert und erzeugen große Datenmengen. Die simulierten Sensordaten müssen üblicherweise für die Weiterverarbeitung oder Sensordatenfusion an den Zentralrechner übermittelt werden, was zu einer sehr hohen Netzwerklast führt. Schlechte Netzwerkverbindungen können daher die Echtzeitfähigkeit eines Systems beeinträchtigen, selbst wenn viele GPUs verfügbar sind. Daher sollte verstärkt auf die Anbindung der Grafikkarten an den Zentralrechner sowie auf die Verteilung der Rechenlast auf mehrere Grafikkarten geachtet werden.

BILD 3 Vier Kamera-RSIs: gewöhnliche Linse vorn (a), hinten (b), rechts (c) und links (d) am Fahrzeug (© IPG Automotive)

ECHTZEITSIMULATION EINES MULTI-SENSOR-SETUPS

In CarMaker simuliert der Hauptsimulationsprozess das Egofahrzeug auf einer Central Processing Unit (CPU), während die physikalischen Sensormodelle von CarMaker in separaten Prozessen simuliert werden. Dies ermöglicht es, die Arbeitslast der Sensorberechnung einfach auf mehrere GPUs zu verteilen. Die verfügbaren GPUs, die auch innerhalb eines Netzwerks verbunden sind, können bequem über eine grafische Benutzeroberfläche in CarMaker hinzugefügt werden.

Die Simulation physikalischer Sensormodelle wird so auf einer GPU ausgeführt und die Ergebnisse für die Weiterverarbeitung zu einer CPU weitergeleitet. Abhängig von der Art des Sensors und der Parametrierung kann eine einzige GPU Berechnungen von drei bis vier RSI-Sensoren durchführen, bevor sie durch die Anbindung der GPU mit der CPU über Computerbusse limitiert wird.

Wie bereits erwähnt, kann ein typisches Sensor-Setup eines autonomen Fahrzeugs der SAE-Level 4 oder 5 über 20 bis 30 Sensoren verfügen. Als Beispiel wird hier ein Setup aus acht Nah-/Fernbereichsradarsensoren, acht Nah-/Fernbereichslidarsensoren, acht Kameras sowie zwölf Ultraschallsensoren für Anwendungen im Bereich Parkassistenz genommen [6]. Ein solches Setup benötigt üblicherweise mindestens acht Grafikkarten, um die Simulation in Echtzeit durchführen zu können. **BILD 2** zeigt die Visualisierung einer Simulation physikalischer Sensormodelle von aktiven Sensoren aus dem genannten Setup.

BILD 4 Vier Kamera-RSIs: Fischaugenobjektiv vorn (a), hinten (b), rechts (c) und links (d) am Fahrzeug (© IPG Automotive)

BILD 3 und **BILD 4** zeigen eine Kamerasimulation desselben Szenarios, wobei **BILD 4** die Simulation mit einem Fischaugenobjektiv zeigt.

Sollte keine große Anzahl von High-End-GPUs verfügbar sein, kann die Parametrierung von RSI-Sensoren reduziert werden (zum Beispiel Kameraauflösung, Anzahl der verfolgten Strahlen), oder die Echtzeitbedingung kann gelockert und die Simulation dennoch durchgeführt werden. Dem Ansatz der Purpose-Driven Fidelity folgend ist es darüber hinaus möglich, ideale oder phänomenologische Sensormodelle zu nutzen, wenn die Berechnung physikalischer Sensormodelle entweder aufgrund von Hardwareeinschränkungen nicht praktikabel oder möglicherweise nicht notwendig ist.

ZUSAMMENFASSUNG UND FAZIT

Die Entwicklung und das Testen von autonomen Fahrzeugen verlangen neue flexible und effiziente Ansätze wie die Nutzung einer leistungsstarken virtuellen Fahrzeugentwicklungsumgebung. Eine solche Simulationsumgebung muss nicht nur realistische Modelle aller relevanten Subsysteme bereitstellen, sondern auch recheneffizient sein, um eine Echtzeitsimulation des autonomen Fahrens mit mehreren physikalischen Sensormodellen zu ermöglichen. Die Recheneffizienz kann durch den Einsatz von Multi-GPU-Parallelisierung erreicht werden.

Die Verwendung einer solchen Simulationsumgebung mit den Vorteilen des Multi-GPU Processing ermöglicht es ebenfalls, automatisch mehrere Testkataloge zu simulieren, mit Milliarden von Kilometern [2], Szenarienvariationen, aber auch seltenen Corner Cases in einer Closed-Loop-Steuerung. So kann der Software-Stack für autonomes Fahren vollständig oder in Teilen getestet werden, um die korrekte Funktion aller beteiligten Subsysteme und letztendlich die Sicherheit eines autonomen Fahrzeugs sicherzustellen.

LITERATURHINWEISE

[1] Costlow, T.: Fusing Sensors for the Automated Driving Future. Tagung SAE Mobilus, 2019. Online: https://saemobilus.sae.org/automated-connected/feature/2019/02/fusing-sensors-for-the-automateddriving-future, aufgerufen am 02.01.2020

[2] Waymo Safety Report: On the Road to Fully Self-Driving, 2018. Online: https://storage.googleapis.com/sdc-prod/v1/safety-report/Safety%20Report%202018.pdf, aufgerufen am 14.05.2020

[3] Herrmann, M.: Sensormodelle für die Entwicklung und Absicherung automatisierter Fahrfunktionen. In: ATZelektronik 14 (2019), Nr. 6, S. 50-53

[4] Yun, Z.; Iskander, M. F.: Ray Tracing for Radio Propagation Modeling: Principles and Applications. In: IEEE Access (2015), Nr. 3, S. 1089-1100

[5] Mcguire, M.; Enderton, E.; Shirley, P.; Luebke, D.: Real-time stochastic rasterization on conventional GPU architectures. In: Proceedings of the Conference on High Performance Graphics, June 2010, S. 173-182

[6] McKinsey & Company: Automotive software and electronics 2030, July 2019. Online: https://www.mckinsey.com/~/media/mckinsey/industries/automotive%20and%20assembly/our%20insights/mapping%20the%20automotive%20software%20and%20electronics%20landscape%20through%202030/automotive-software-and-electronics-2030-final.ashx, aufgerufen am 14.05.2020