DOI: 10.2195/lj_Proc_thiel_de_201811_01
URN: urn:nbn:de:0009-14-47541

Deep-Learning-Verfahren zur 3D-Objekterkennung in der Logistik

Deep learning for 3D object recognition in logistics

Marko Thiel
Johannes Hinckeldeyn
Jochen Kreutzfeldt

Institut für Technische Logistik
Technische Universität Hamburg

Die zuverlässige Erkennung von Objekten in Sensordaten ist Grundvoraussetzung für die Autonomisierung logistischer Prozesse. Insbesondere die Erkennung von Objekten in 3D-Sensordaten ist für flexible autonome Anwendungen wichtig. Für die Objekterkennung in 2D-Bilddaten stellen auf neuronalen Netzen basierenden Deep-Learning-Verfahren den Stand der Technik dar. Dieser Beitrag diskutiert verschiedene aktuelle Ansätze, Deep-Learning-Verfahren auch für die 3D-Objekterkennung zu nutzen. Wesentliches Merkmal dieser Ansätze ist die Verwendung von Punktwolken als Eingangsdaten, gegebenenfalls nach vorheriger Segmentierung oder Umwandlung in Voxelgitter. Beispielhafte Anwendungen in der Logistik sind autonome Flurförderzeuge und Kommissionierroboter. Herausforderungen für einen Einsatz bestehen in fehlenden Trainingsdaten, hohen erforderlichen Rechenleistungen für Echtzeitanwendungen und einer noch nicht ausreichenden Erkennungsgenauigkeit.

[Schlüsselwörter: autonome Systeme, 3D-Objekterkennung, Deep-Learning, Punktwolke]

Abstract: The reliable detection of objects in sensor data is a fundamental requirement for the autonomization of logistic processes. Especially the recognition of objects in 3D sensor data is important for flexible autonomous applications. Deep learning represents the state of the art for object recognition in 2D image data. This article presents various current approaches to use deep learning for 3D object recognition. An essential feature of these approaches is the use of point clouds as input data, possibly after prior segmentation or conversion into voxel grids. Examples of applications in logistics are autonomous guided vehicles and order picking robots. The challenges for an application are a lack of training data, high computing requirements for real-time applications and an accuracy that is not yet sufficient.

[Keywords: autonomous systems, 3D object recognition, deep learning, point cloud]

1 EINLEITUNG

In der Logistik finden verschiedene technische Systeme Einsatz, die Menschen bei physischen Tätigkeiten unterstützen oder selbstständig physische Tätigkeiten ausführen. Beispiele solcher Systeme sind fahrerlose Flurförderzeuge sowie Roboter zur Kommissionierung: Fahrerlose Flurförderzeuge führen automatisiert innerbetriebliche Transporte aus und übernehmen Ein- und Auslagervorgänge in Palettenregalen [Jung18; Lind18; Stil18]. Auch für Transporte außerhalb von Logistikumgebungen zur Überbrückung der letzten Meile werden bereits hochautomatisierte Fahrzeuge erprobt [Star18]. Stationäre Kommissionierroboter greifen Artikel aus bereitgestellten, sortenreinen Behältern und stellen automatisiert Kundenaufträge zusammen [Knap18; Righ18]. Mobile Roboter zur Kommissionierung vereinen beide vorgestellten Systeme: Einrichtungen zur Entnahme von Artikeln aus Regalen werden mit fahrbaren Basen kombiniert, damit sich Kommissionierroboter frei durch Lager zur Ware bewegen können [Maga18]. Bereits jetzt könne also viele Kernaufgaben der Logistik automatisiert abgewickelt werden.

Es ist anzunehmen, dass die Nutzung intelligenter Robotik-Systeme in der Logistik in den nächsten Jahren weiter zunehmen wird. Arbeitskräfte werden durch Roboter unterstützt oder kollaborieren direkt mit diesen [Logi18, S.46 f.]. Autonome Flurförderzeuge als Basis einer adaptiven Logistik ermöglichen den Aufbau wandlungsfähiger Fabriken [Fach18, S.59]. Je enger autonome Robotik-Systeme kollaborativ eingesetzt und je flexibler deren Aufgaben ausgestaltet werden, desto wichtiger wird die Fähigkeit, das eigene Umfeld erfassen zu können.

Ein häufig als intelligent, selbstständig oder flexibel bezeichnetes Verhalten autonomer Systeme ist mit den aktuell verwendeten Sensoren und Algorithmen jedoch nur eingeschränkt zu realisieren. Die in fahrerlosen Flurförderzeugen verbauten 2D-Laserscanner gewährleisten den Personenschutz nach DIN EN 1525 [Dine97], ermöglichen aber keine genaue Erkennung und Klassifizierung von Objekten und Hindernissen. Gerade diese Funktion auf

Article is protected by German copyright law

DOI: 10.2195/lj_Proc_thiel_de_201811_01
URN: urn:nbn:de:0009-14-47541

Grundlage von 3D-Sensoren wird für autonome Fahrzeuge benötigt [Ullr18]. Bezüglich der Verwendung von Kommissionierrobotern konnte am Beispiel der Amazon Robotics Challenge [Amaz17] gezeigt werden, dass für die Kommissionierung heterogener Artikel grundsätzliche Lösungsvorschläge bestehen. Für einen industriellen Einsatz ist die Robustheit solcher Systeme noch nicht ausreichend [DzMF18]. Hierfür stellt die Nutzung der verfügbaren 3D-Sensorik zur Objekterkennung eine mögliche Lösung dar.

Bei der 2D-Objekterkennung stellen Deep-Learning-Verfahren den aktuellen Stand der Technik dar. Ziel dieses Beitrags ist es, verschiedene Ansätze vorzustellen, Deep-Learning-Verfahren auch zur 3D-Objekterkennung zu nutzen.

2 GRUNDLAGEN

In den folgenden beiden Abschnitten soll zum einen geklärt werden, was genau unter Objekterkennung in Sensordaten verstanden wird. Zum anderen erfolgt eine kurze Einführung in das Thema neuronale Netze bzw. Deep-Learning.

2.1 OBJEKTERKENNUNG

Der Begriff „Objekterkennung“ umfasst mehrere konkrete Problemstellungen der maschinellen Bildverarbeitung, die sich grundsätzlich in folgende drei Bereiche einteilen lassen [SüRo14, S.589]:

- Objektklassifizierung: Das auf einem Bild oder allgemein in Sensordaten dargestellte Objekt wird einer vorher eingelernten Klasse/Kategorie zugeordnet. Beispiele: *Person*, *Gabelstapler*, *Palette*.

- Objektlokalisierung: Zusätzlich zur Objektklassifizierung erfolgt eine Bestimmung des Ortes gefundener Objekte in den Sensordaten. Die gefundene Position wird durch ein umschließendes Rechteck (Bounding Box) beschrieben und hervorgehoben.

- Semantische Segmentierung: Jedes Element der Sensordaten wird einer Klasse/Kategorie zugeordnet. Liegt ein 2D-Bild vor, erhält jedes einzelne Pixel eine zugehörige Klassen-Annotation.

2.2 DEEP-LEARNING-VERFAHREN

Deep-Learning ist ein auf künstlichen neuronalen Netzen basierendes Verfahren des maschinellen Lernens. Generelles Ziel ist, eine nichtlineare Abbildung von Eingangsdaten auf Ausgangsdaten zu lernen. Bei der Klassifizierung von 2D-Bildern besteht diese Abbildung z.B. darin, Eingangsdaten in Form eines Bildes einer diskreten Klasse - der des erkannten Objektes - zuzuordnen. Das Lernen erfolgt datengetrieben, also nur auf Grundlage bereitgestellter Trainingsdaten. Diese Trainingsdaten enthalten eine Vielzahl von Eingangsdaten mit dazugehörigen Ausgangsdaten bzw. den korrekten Ergebnissen der zu lernenden Zuordnung. Im einfachsten Fall besteht ein künstliches neuronales Netzwerk aus einer einzigen sogenannten Schicht: Zuerst werden die Eingangsdaten anhand einzelner Parameter linear kombiniert, danach wird eine nichtlineare Funktion auf die so erzeugten Werte angewandt. Tiefe (*deep*) neuronale Netze setzen sich aus mehreren hintereinander angeordneten Schichten zusammen. Der Prozess des Lernens/Trainierens besteht nun darin, die Parameter des Netzwerks so anzupassen, dass eine korrekte Abbildung für alle Eingangsdaten erzielt wird. Während des Trainings werden folgende Schritte durchlaufen: 1) Ein Trainingsbeispiel wird vorwärts durch das Netzwerk geschleust. 2) Die Abweichung von der zu erreichenden, korrekten Lösung wird mithilfe einer Kostenfunktion quantifiziert. 3) Durch Rückrechnen kann der Einfluss jedes Parameters auf das Ergebnis bestimmt werden. 4) Alle Parameter werden entsprechend ihres Einflusses angepasst, sodass sich die Abweichung zwischen korrektem Ergebnis laut Trainingsdaten und dem anhand der aktuellen Parameter errechneten Ergebnis verringert. Damit ist ein Trainings-Schritt abgeschlossen. Dieser Vorgang wird wiederholt, bis ein gewünschter Trainingszustand des Netzwerks erreicht ist. Weitere Informationen dazu finden sich unter anderem in [GoBC16].

Als Erweiterung der neuronalen Netze und insbesondere gebräuchlich bei der Klassifizierung von Bilddaten bzw. allgemein strukturierten Daten sind *Convolutional-Neural-Networks (CNN).* Diese nutzen die für die Semantik eines Bildes wichtige lokale Struktur von Pixeldaten: Die Ausgansdaten einer Schicht werden durch Faltung der Eingangsdaten mit rechteckigen Filtermatrizen erzeugt, die dafür schrittweise über die Eingangsdaten geführt werden. Filtermatrizen sind hierbei quadratische Arrays, deren Zellen die Parameter enthalten. Eine Faltung entspricht der elementweisen Multiplikation mit anschließender Addition der Teilergebnisse. In der ersten Schicht bestehen die Eingangsdaten aus dem zu klassifizierendem Bild in ursprünglicher Form des Pixelrasters. Für weitere Informationen sei wieder verwiesen auf [GoBC16].

Ein großer Vorteil von Deep-Learning-Verfahren ist, dass Repräsentationen der Daten, die *Features,* im Prozess des Trainings erlernt werden. Dieses geschieht automatisch in den einzelnen Schichten, ohne eine manuelle Auswahl tätigen zu müssen oder vor Beginn des Lernens Features aus den Daten zu extrahieren [LeBH15, S.438].

3 DATENSÄTZE

Für Training und Test aber auch den Vergleich von Algorithmen werden Sammlungen von annotierten Trainingsdaten benötigt. Dies sind Datensätze, für die die korrekte Lösung der Erkennungsaufgabe bekannt und mit angegeben ist. Im Fall der Objektlokalisierung sind also Klasse und Geometrie der Bounding Box für jedes Beispiel

Article is protected by German copyright law

DOI: 10.2195/lj_Proc_thiel_de_201811_01
URN: urn:nbn:de:0009-14-47541

in den Daten enthalten. Es stehen verschiedene offene Trainingsdatensätze zur Verfügung, von denen hier drei Beispiele präsentiert werden:

- KITTI Vision Benchmark Suite [GeLU12]: Datensatz und Benchmark aus dem Bereich des voll-/hochautomatisierten Fahrens. Verkehrswege und Objekte des Straßenverkehrs wurden mit unterschiedlichen Sensoren (3D-Laserscanner, Stereo-Kamera, 2D-Kamera) aufgezeichnet und anschließend annotiert (z.B. *PKW* oder *Fußgänger*). Die Benchmarks sind in einzelne Kategorien aufgeteilt, unter anderem *3D object detection.*

- ModelNet [ZSKF15] [Prin18]: Die bereitgestellten Datensätze bestehend aus verschiedenen manuell annotierten CAD-Modellen (z.B. *Stuhl* oder *Flugzeug*). Der Datensatz ModelNet 40 enthält Modelle von 40 verschiedenen Klassen. Da die Daten als CAD-Modell vorliegen, müssen diese vor Verwendung als Trainingsdaten gegebenenfalls in Voxelgitter oder Punktwolken umgewandelt werden.

- Semantic3D.net Large-Scale Point Cloud Classification Benchmark [HSLW17]: Datensatz aus annotierten Punktwolken mehrerer natürlicher Außen-Umgebungen bzw. städtischer Szenen (Objekte sind z.B. *Gebäude*, *Vegetation* oder *PKW*).

Spezifische Datensätze mit Objekten aus der Logistik sind zum jetzigen Zeitpunkt nicht bekannt.

Als Metrik zur Bewertung der auf einem Datensatz erzielten Erkennungsergebnisse wird in vielen Fällen sowohl die relative Anzahl der richtig vorhergesagten Objekt-Klassen herangezogen als auch die Anzahl erreichter Mindest-Überdeckungen der bestimmten Bounding Boxen mit den wahren Bounding Boxen (siehe z.B. [GeLU12]).

4 Sensoren und Datenformate

Sowohl für den Betrieb von 3D-Objekterkennungs-Systemen als auch für die vorherige Aufnahme von Trainingsdaten werden Sensoren benötigt, welche die Umgebung dreidimensional vermessen können. 3D-Sensoren liefern Tiefen- bzw. Entfernungsinformationen umgebender Objekte ausgehend vom Sensorstandort. Für den industriellen Einsatz sind 3D-Sensoren verschiedener Hersteller kommerziell erhältlich. Folgend werden drei verbreitete Technologieprinzipien vorgestellt (siehe auch [Oste17]):

- Stereo-Kamera: Basierend auf zwei 2D-Kameras, die um eine Basisbreite versetzt voneinander montiert sind, können pixelweise Entfernungswerte durch Triangulation der Einzelmessungen berechnet werden. (Beispiele: SICK Visionary-B [Sick18a], Roboception rc_visard [Robo18])

- ToF-Kamera: *Time-of-Flight*-Kameras arbeiten nach dem Lichtlaufzeitverfahren: Ausgehend von einem Pixelraster wird pixelweise die Laufzeit ausgesandten und reflektierten Infrarotlichts bestimmt, um daraus die Entfernung der Reflexionsflächen zu berechnen. Im Vergleich zu 3D-Laserscannern erfolgt die Entfernungsmessung aller Punkte zeitgleich. Die Ausgabe des Sensors wird auch als 2,5D-Tiefenbild bezeichnet. (Beispiele: SICK Visionary-T [Sick18b], Basler ToF-Kamera [Basl18], ifm electronic O3Dxxx-Serie [Ifme18])

- 3D-Laserscanner (auch 3D-LiDAR): Analog zur Funktionsweise klassischer 2D-Laserscanner arbeiten 3D-Laserscanner mit rotierenden Lasern, welche punktweise die Umgebung abtasten. Entweder wird ein einzelner Laser in zwei Raumrichtungen rotiert oder mehrere Laser in einer Ebene. Die Entfernung jedes einzelnen Messpunktes wird über die Laufzeit der Reflektion bestimmt. (Beispiel: Velodyne [Velo18]). Eine neuere Entwicklung im Bereich der Laserscanner sind Solid-State-LiDAR-Sensoren ohne bewegliche Bauteile (Beispiel: Quanergy S3 [Quan18]).

Ein in 3D-Sensorik-Anwendungen häufig genutztes Datenformat ist die Punktwolke. Hierbei wird jeder Messpunkt durch drei Raumkoordinaten charakterisiert. Die Daten werden entweder bereits vom Sensor als Punktwolke ausgegeben oder lassen sich leicht in das Format einer Punktwolke umwandeln [Poin18; Robo17, S.525–530]. Für diese und weitere Operationen existieren umfangreiche Open-Source-Bibliotheken, wobei insbesondere die *Point Cloud Library (PCL)* zu nennen ist [RuCo11].

5 Verfahren zur 3D-Objekterkennung

Klassische Verfahren der 3D-Objekterkennung basieren auf einer vorherigen Generierung von Features, also einer numerischen, abstrakten Repräsentation der Trainings- und Eingangsdaten. Dabei steht eine große Anzahl möglicher Features zur Verfügung, aus denen eine für den Anwendungszweck geeignete Auswahl getroffen werden muss [Alex12]. Ein großer Vorteil der Verwendung von Deep-Learning-Verfahren besteht darin, dass die Features auf Grundlage der Trainingsdaten selbst erlernt werden. Folgend werden verschiedene aktuelle Ansätze vorgestellt, Deep-Learning-Verfahren für die 3D-Objekterkennung zu nutzen. Der Fokus liegt dabei auf einer Darstellung grundlegender Ideen und Prinzipien anhand ausgewählter Beispiele.

5.1 Direkte Verwendung von Punktwolken

Die in der 2D-Objekterkennung fast ausschließlich verwendeten Convolutional-Neural-Networks nutzen die semantisch wichtige Struktur der gegebenen Eingangsdaten. Das Vertauschen der Pixel eines Bildes würde die da-

Article is protected by German copyright law

DOI: 10.2195/lj_Proc_thiel_de_201811_01
URN: urn:nbn:de:0009-14-47541

hinterliegende semantische Information eliminieren. Anders ausgedrückt: Werden Bildpunkte eines Fotos vertausch, ist der abgebildete Gegenstand nicht mehr zu erkennen. Punktwolken dagegen bestehen aus einer Menge einzelner Punkte in einem Koordinatenraum. Jeder Punkt ist durch drei Koordinaten bezüglich eines gemeinsamen Ursprungs bestimmt. Eine Veränderung der Reihenfolge der Punkte in einer Menge ändert die Semantik der Punktwolke nicht, die Darstellung ist permutationsinvariant bezüglich der einzelnen Punkte.

Eine Lösung zur Nutzung von Deep-Learning-Verfahren trotz dieser Permutationsinvarianz liegt in der Verwendung symmetrischer Funktionen. Symmetrische Funktionen sind invariant bezüglich der Reihenfolge ihrer Argumente (einfache Beispiele sind die Addition und die Multiplikation). Grundidee ist, die Elemente der Punktewolke in den ersten Schichten des neuronalen Netzwerks zunächst getrennt voneinander zu transformieren. Auf die Ausgabe dieser Schichten werden dann symmetrische Funktionen angewandt (z.B. Max-Pooling – Anwendung der Maximum-Funktion auf Teilbereiche der Daten). Mit solchen Netzwerkarchitekturen können bereits Klassifizierungssaufgaben gelöst werden. Auch eine semantische Segmentierung lässt sich durchführen. Dazu wird ein weiteres, paralleles neuronales Netzwerk verwendet, dass die zur Klassifikation nötigen, erlernten globalen Features mit lokalen Features zusammenführt und Ausgaben für jedes Element der Punktwolke erzeugt [CSKG17]. Aufbauend auf der vorgestellten Arbeit konnten weitere Verbesserungen durch die Nutzung hierarchischer Features erzielt werden: Strukturen sowie zusammengehörige Punkte werden dadurch besser erfasst [QYSG17].

5.2 Konvertierung der Punktwolke in Voxel

Convolutional-Neural-Networks arbeiten auf strukturierten Daten, beispielsweise auf Pixelrastern (2D-Bilder). Analog ist eine Verarbeitung von 3D-Rastern, sogenannten Voxelgittern, möglich. Voxel stellen eine 3D-Erweiterung von Pixel dar. Um aus Punktwolken Voxelgitter zu erzeugen, erfolgt zunächst eine Diskretisierung des Koordinatenraums. Daraufhin werden einzelne Voxel des Gitters als belegt markiert, wenn in diesen ein Element oder mehrere Elemente der ursprünglichen Punktwolke zu finden sind. Auf diesem Voxelgitter aus belegten und freien Gitterzellen können nun 3D-Filtermatrizen zur Faltung verwendet werden, entsprechen der 2D-Faltung im Zusammenhang mit 2D-Bildern. Mithilfe dieses Ansatzes lassen sich bereits auf einzelnen Segmenten einer Punktwolke Klassifizierungsaufgaben durchführen [MaSc15]. In Kombination mit *Region-Proposal-Networks,* Netzwerkarchitekturen zur Vorhersage von Bounding Boxes, ist selbst auf größeren Datenmengen die Lokalisierung von Objekten umsetzbar [ZhTu17].

5.3 Generierung von 2D-Ansichten

Die Objekterkennung in 2D-Bilddaten ist Stand der Technik. Kernidee der meist *Multi-View* genannten Verfahren ist daher, das 3D-Erkennungsproblem in ein 2D-Erkennungsproblem zu überführen. Dazu werden die zu verarbeitenden 3D-Objekte von mehreren Seiten als 2D-Bilder gerendert bzw. auf 2D-Ebenen projiziert. Die daraus entstehenden 2D-Repräsentationen in Form von Bildern können dann als Eingangsdaten für klassische Convolutional-Neural-Networks verwendet werden. Nach diesem Prinzip arbeitende Ansätze unterscheiden sich unter anderem darin, wie mit den gewonnen 2D-Bildern weiter verfahren wird. Optionen dafür bestehen in speziellen Schichten innerhalb der neuronalen Netze, welche die Informationen der vielen einzeln verarbeiteten Bilder zusammenführen [SMKL15] oder aber in Netzwerkarchitekturen, die mit einer deutlich geringeren Anzahl gerenderter Ansichten auskommen [KaMN16].

In den referenzierten Veröffentlichungen wurden einzelne Objekte nur klassifiziert, nicht innerhalb einer größeren Menge von Daten lokalisiert. Grundsätzlich ließen sich mithilfe der 2D-Ansichten auch Lokalisierungsaufgaben ausführen. Eine andere Alternative besteht in der vorherigen Segmentierung der Punktwolke.

5.4 Nutzung von RGB-D Daten

Neben den vorgestellten Ansätzen, 3D-Objekterkennungsprobleme auf Basis von Punktwolke zu lösen, existieren weitere Verfahren, die neben den Tiefeninformationen (Depth) eines 3D-Sensors auf RGB-Daten einer 2D-Kamera zugreifen [QLWS17]. Aufgrund der zusätzlich benötigten Sensorik wird an dieser Stelle auf eine weitere Betrachtung verzichtet.

6 Ableitung des Forschungsbedarfs

Deep-Learning-Verfahren belegen in 3D-Objekterkennungs-Benchmarks auf offenen Datensätzen die ersten Positionen [Prin18]. Ein großer Vorteil dieser Verfahren, gerade auch im Vergleich zu früheren Lösungen, liegt im automatischen Erlernen der Features. Es ist daher sinnvoll zu prüfen ob, und wenn ja, wie sich diese Ansätze in der Logistik nutzen lassen. Dabei werden zwei wesentliche Richtungen möglicher Forschungen vorgeschlagen: Datensätze und Algorithmen.

Obwohl verschiedene freie Datensätze zum Training und Test von 3D-Objekterkennungsalgorithmen zur Verfügung stehen, ist zum jetzigen Zeitpunkt kein spezifischer Datensatz von Objekten aus der Logistik bekannt. Um Anwendungsmöglichkeiten im Umfeld der Logistik praxisorientiert zu erproben, bedarf es einer Erstellung eigener Datensätze. Nicht nur zu Testzwecken, sondern auch zur Entwicklung konkreter Produkte, werden diese Datensätze zwingend benötigt.

Article is protected by German copyright law

DOI: 10.2195/lj_Proc_thiel_de_201811_01
URN: urn:nbn:de:0009-14-47541

Wie am Beispiel von ModelNet gezeigt wurde, werden auch auf CAD-Modellen basierende Daten für das Training und den Vergleich von Algorithmen genutzt. Hierin besteht eventuell eine Lösung, in kurzer Zeit eine größere Menge Trainingsdaten für Anwendungen in der Logistik zu generieren. Es bleibt zu prüfen, welchen Einfluss die Art der Trainingsdaten (real/synthetisch) auf einen Einsatz in realen Umgebungen hat.

Die mithilfe von Deep-Learning-Verfahren erreichten Genauigkeiten bei der Klassifizierung und Lokalisierung von Objekten scheinen für einen Einsatz in der Praxis noch zu gering. Im ModelNet-Benchmark (Klassifizierung) erreichen mehrere Deep-Learning-Verfahren eine maximale Genauigkeit von über 90 %, aber noch unter 96 % [Prin18]. Bei der Unteraufgabe der Objektlokalisierung des KITTI-Benchmarks liegt der maximale Wert bei 73,66 % [Kitt18]. Empfehlenswert ist ein Test der Algorithmen mit Trainingsdaten der konkret zu lösenden Problemstellung der Logistik. Ein eingeschränkter Umfang der Objekterkennungsaufgabe führt gegebenenfalls zu höheren Genauigkeiten.

Eine Bewertung der Echtzeitfähigkeit für den praktischen Einsatz auf Grundlage berichteter Laufzeiten scheint problematisch. Zum einen sind Laufzeitangaben nicht in allen Benchmarks abrufbar, zum anderen hängt die Laufzeit stark von der eingesetzten Hardware ab (Grafikkarte, Prozessor). Um verschiedene Algorithmen zu vergleichen und die Laufzeit bewerten zu können, sollten eigene Versuche mit identischer Hardware durchgeführt werden.

Einfluss auf die Laufzeit der Algorithmen aber auch auf die Auswahl geeigneter Sensoren hat die für das zu lösende Erkennungsproblem benötigte Auflösung. Hierfür ist zu testen, welche Auflösungen für konkrete Problemstellungen notwendig sind, beispielsweise für die Erkennung typischer Objekte in einer Lagerhallte (Personen, Paletten, Regale, Flurförderzeuge). Je geringer die Auflösung gewählt werden kann, desto geringer ist die benötigte Rechenleistung bzw. bei gleicher Rechenleistung die benötigte Zeit.

7 ANWENDUNGEN IN DER LOGISTIK

Im Kontext der Logistik sind insbesondere zwei Bereiche der Objekterkennung von Interesse: Objektlokalisierung und semantische Segmentierung. Alle Anwendungen, die das Erkennen von Objekten und die Bestimmung der dazugehörigen Positionen erfordern, könnten potentiell von einer 3D-Objekterkennung profitieren. Vorteilhaft dabei wirkt sich gegebenenfalls auch die hohe Standardisierung der in diesen Umgebungen vorkommenden Objekte aus. Folgend werden mögliche Anwendungen im Umfeld der Logistik dargestellt:

- **Anwendung**: Fahrerlose Flurförderzeuge
    - Freie Fahrwege werden automatisch erkannt und nicht nur auf Basis einer hochgenauen Referenzkarte bestimmt.
    - Hindernisse werden erkannt und lokalisiert. Auf Grundlage der Klassifizierung wird entschieden, ob statische oder dynamische Objekte den Weg blockieren und Ausweichmaßnahmen sinnvoll sind. Weiterhin wird die 3D-Sensorik zur Detektion von Hindernissen genutzt, die in beliebiger Höhe den Fahrweg versperren (siehe auch [Ullr18]).
    - Die Position und Orientierung zu transportierenden Ladungsträger wird durch die Sensorik erkannt. Damit können auch gegenüber der ursprünglichen Sollposition verschobene Ladungsträger aufgenommen werden.
    - Erkannte statische Objekte dienen als Orientierungspunkte zur Ortung des Fahrzeugs.
- **Anwendung**: Kommissionierroboter (stationär und mobil)
    - Beim Kommissionieren aus Behältern werden 3D-Sensoren nicht nur zur Unterstützung bei der Lokalisierung von Objekten und Greifpositionen genutzt, sondern auch zur Klassifizierung. Als Zusatz zur 2D-Klassifizierung erhöht dies die Robustheit des Gesamtsystems (siehe auch [SDBU17]).
    - Beim Greifen von Artikel aus dem Regal wird die genaue Position des Artikels bestimmt und ggf. überprüft, ob es sich um den korrekten Artikel handelt.
- **Anwendung:** Assistenzsysteme für Flurförderzeuge
    - Personen und Hindernisse werden zuverlässig erkannt, eine Warnung an die Fahrer ausgegeben bzw. eine automatische Bremsung des Fahrzeugs eingeleitet (siehe z.B. auch [LaJo17]).
- **Anwendung**: Assistenzsysteme für manuelle Kommissionierung
    - Für einen Kunden manuell kommissionierte Bestellungen werden auf Vollständigkeit und falsche Artikel geprüft (siehe auch [HKGA17]).
- **Anwendung:** Kollaborative Roboter
    - Neben einer Absicherung der Roboter gegen Kollisionen (z.B. Drehmomentmessung in

Article is protected by German copyright law

DOI: 10.2195/lj_Proc_thiel_de_201811_01
URN: urn:nbn:de:0009-14-47541

den Gelenken) werden Menschen und Hindernisse aktiv auf Grundlage von 3D-Sensordaten erkannt.

  - Beim Wechseln von Arbeitsstationen werden Abweichungen zum Sollstandort des Roboters erkannt und eingelernte Trajektorien und Prozesse automatisiert korrigiert.

- **Anwendung:** Erstellung semantischer Karten

  - Objekte in aufgenommene Punktwolken von Logistik-und Produktionsumgebungen werden automatisiert Lokalisiert und mit Annotationen versehen. Ein manuelle Nachbearbeitung, um die Punktwolke mit semantischen Informationen zu versehen, entfällt (für 2D-Karten siehe z.B. [HiMa17]).

- **Anwendung:** Augmented Reality

  - Reale Objekte werden lokalisiert und mit dazugehörigen, virtuellen Beschriftungen versehen. Anwendungsbereiche finden sich sowohl in der Einarbeitung von neuen Mitarbeitern als auch bei der Remote-Unterstützung von Service-Technikern im Feld bzw. vor Ort beim Kunden.

8 Zusammenfassung und Fazit

Die Automatisierung von Prozessen der Logistik ist Stand der Technik und wird in den nächsten Jahren weiter voranschreiten. Eine Forderung nach intelligenterem Verhalten und zunehmender Flexibilität setzt Sensoren und Algorithmen voraus, die die zuverlässige Klassifizierung und Lokalisierung von 3D-Objekten erlauben. 3D-Sensoren, deren Ausgabedaten als Punktwolke dargestellt werden können, sind seit einiger Zeit kommerziell erhältlich. Bezüglich der Algorithmen wurden insbesondere in den letzten Jahren verschiedene Verfahren vorgestellt, welche die im 2D-Bereich den Stand der Technik darstellenden Deep-Learning-Verfahren auch auf 3D-Daten in Form von Punktwolken anwenden. Ansätze sind dabei z.B. die Umwandlung des 3D-Erkennungsproblems in ein 2D-Erkennungsproblem durch Projektionen der Objekte, Rasterung der Punktwolke als Voxel zur Verwendung mit 3D-Faltungen oder die direkte Verwendung von Punktwolken als Eingangsdaten mithilfe symmetrischer Funktionen. Noch scheinen die erreichten Erkennungsgenauigkeiten und erforderlichen Rechenleistungen für einen generellen Echtzeiteinsatz in der Logistik nicht zu genügen. Auch stehen dafür zurzeit keine erforderlichen Trainingsdatensätze zur Verfügung. Sobald dies aber der Fall ist, finden sich viele mögliche Anwendungen einer 3D-Objekterkennung im Kontext autonomer Flurförderzeuge, stationärer und mobiler Kommissionierroboter, Assistenzsysteme für Flurförderzeuge, Erstellung semantischer Karten sowie VR/AR. Es wird empfohlen, konkrete Teilprobleme von Erkennungsaufgaben der Logistik mit selbst erstellten Trainingsdaten unter Nutzung verschiedener Algorithmen zu testen und zu evaluieren.

Literatur

[Alex12] Alexandre, Luís A.: 3D Descriptors for Object and Category Recognition: a Comparative Evaluation. In: *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vilamoura, Portugal, 2012

[Amaz17] Amazon Robotics: *2017 Amazon Robotics Challenge Official Rules*. URL https://www.amazonrobotics.com/site/binaries/content/assets/amazonrobotics/arc/2017-amazon-robotics-challenge-rules-v3.pdf. - abgerufen am 2018-08-01

[Basl18] Basler AG: *Basler Time-of-Flight-Kamera*. URL https://www.baslerweb.com/de/produkte/kameras/3d-kameras/time-of-flight-kamera/. - abgerufen am 2018-08-01

[CSKG17] Charles, R. Qi; Su, Hao; Kaichun, Mo; Guibas, Leonidas J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: : IEEE, 2017 — ISBN 978-1-5386-0457-1, S. 77–85

[Dine97] DIN EN 1525: Sicherheit von Flurförderzeugen - Fahrerlose Flurförderzeuge und ihre Systeme, Beuth Verlag GmbH (1997)

[DzMF18] Dziedzitz, Jonathan; Markert, Kai; Furmans, Kai: Fortschritte bei der automatischen Kommissionierung m Beispiel der Amazon Robotics Challenge. In: , *VDI-Berichte*. Bd. 2325. München, 2018

[Fach18] Fachforum Autonome Systeme im Hightech-Forum: Autonome Systeme - Chancen und Risiken für Wirtschaft, Wissenschaft und Gesellschaft (Abschlussbericht Langerversion) (2018)

[GeLU12] Geiger, Andreas; Lenz, Philip; Urtasun, Raquel: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: *Conference on*

Article is protected by German copyright law

Computer Vision and Pattern Recognition (CVPR), 2012

[GoBC16] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron: *Deep Learning* : MIT Press, 2016

[HiMa17] Himstedt, Marian; Maehle, Erik: Online semantic mapping of logistic environments using RGB-D cameras. In: *International Journal of Advanced Robotic Systems* Bd. 14 (2017), Nr. 4, S. 172988141772078

[HKGA17] Hochstein, Maximilian; Kunert, Christoph; Glöckle, Johannes; Averweg, Manuel; Weil, Hendrik; Furmans, Kai: Konsolidierassistent – Assistenzsystem für manuelle Konsolidier- und Sortierprozesse in Distributionszentren. In: *Logistics Journal: Proceedings* Bd. 2017 (2017), Nr. 10

[HSLW17] Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J. D.; Schindler, K.; Pollefeys, M.: SemanticC3D.net: A new large-scale point cloud classification benchmark. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* Bd. IV-1/W1 (2017), S. 91–98

[Ifme18] ifm electronic GmbH: *3D-Kameras*. URL https://www.ifm.com/de/de/category/020/020_030. - abgerufen am 2018-08-01

[Jung18] Jungheinrich AG: *Automatisierte Flurförderzeuge*. URL https://www.jungheinrich.de/produkte/automatisierte-komponenten/fahrerlose-transportfahrzeuge/automatisierte-flurfoerderzeuge. - abgerufen am 2018-08-01

[KaMN16] Kanezaki, Asako; Matsushita, Yasuyuki; Nishida, Yoshifumi: RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints. In: *arXiv:1603.06208 [cs]* (2016).

[Kitt18] The KITTI Vision Benchmark Suite: *3D Object Detection Evaluation 2017*. URL http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d. - abgerufen am 2018-08-01

[Knap18] Knapp AG: *Kommissionieren - Pick-it-Easy Robot*. URL https://www.knapp.com/loesungen/technologien/kommissionieren/. - abgerufen am 2018-08-01

[LaJo17] Lang, Armin; Johannes, Fottner: Konzeption eines kamerabasierten Kollisionswarnsystems zur Prävention von Arbeitsunfällen an Gabelstaplern. In: *Logistics Journal: Proceedings* Bd. 2017 (2017), Nr. 10

[LeBH15] LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey: Deep learning. In: *Nature* Bd. 521 (2015), Nr. 7553, S. 436–444

[Lind18] Linde Material Handling GmbH: *Linde Automatisierungslösungen*. URL https://www.linde-mh.de/de/Loesungen/Automatisierung/. - abgerufen am 2018-08-01

[Logi18] *Logistics Trend Radar - Delivering insight today, creating value tomorrow.* (Version 2018/2019) : DHL Trend Research, 2018

[Maga18] Magazino GmbH: *Magazino Produkte - Übersicht über alle Roboter und Lösungen*. URL https://www.magazino.eu/toru/. - abgerufen am 2018-08-01

[MaSc15] Maturana, Daniel; Scherer, Sebastian: VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In: IEEE, 2015 - ISBN 978-1-4799-9994-1, S. 922–928

[Oste17] Osterwood, Chris: How to Choose a 3D Vision Technology. In: *ROSCon*. Vancouver, 2017

[Poin18] Point Cloud Library (PCL): *Module io*. URL http://docs.pointclouds.org/trunk/group__io.html. - abgerufen am 2018-08-01

[Prin18] *Princeton ModelNet*. URL http://modelnet.cs.princeton.edu/. - abgerufen am 2018-08-01

[QLWS17] Qi, Charles R.; Liu, Wei; Wu, Chenxia; Su, Hao; Guibas, Leonidas J.:

Article is protected by German copyright law

Frustum PointNets for 3D Object Detection from RGB-D Data. In: *arXiv:1711.08488 [cs]* (2017)

[Quan18] Quanergy Systems, Inc.: *Products – Quanergy*. URL https://quanergy.com/products/. - abgerufen am 2018-08-01

[QYSG17] Qi, Charles Ruizhongtai; Yi, Li; Su, Hao; Guibas, Leonidas J: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R. (Hrsg.): *Advances in Neural Information Processing Systems 30* : Curran Associates, Inc., 2017, S. 5099–5108

[Righ18] *RightHand Robotics*. URL https://www.righthandrobotics.com. - abgerufen am 2018-08-01

[Robo17] *Robot operating system (ROS)*. New York, NY : Springer Berlin Heidelberg, 2017 - ISBN 978-3-319-54926-2

[Robo18] Roboception GmbH: *rc_visard 3D Sensor*. URL https://roboception.com/de/rc_visard/. - abgerufen am 2018-08-01

[RuCo11] Rusu, Radu Bogdan; Cousins, Steve: 3D is here: Point Cloud Library (PCL). In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, 2011

[SDBU17] Schwäke, Kim; Dick, Ilja; Bruns, Rainer; Ulrich, Stephan: Entwicklung eines flexiblen, vollautomatischen Kommissionierroboters. In: *Logistics Journal: Proceedings* Bd. 2017 (2017), Nr. 10

[Sick18a] SICK AG: *Visionary-B*. URL https://www.sick.com/de/de/vision/3d-vision/visionary-b/c/g348851. - abgerufen am 2018-08-01

[Sick18b] SICK AG: *3D Visionary-T*. URL https://www.sick.com/de/de/vision/3d-vision/visionary-t/c/g358152. - abgerufen am 2018-08-01

[SMKL15] Su, Hang; Maji, Subhransu; Kalogerakis, Evangelos; Learned-Miller, Erik: Multi-view Convolutional Neural Networks for 3D Shape Recognition. In: *2015 IEEE International Conference on Computer Vision (ICCV)* : IEEE, 2015 - ISBN 978-1-4673-8391-2, S. 945–953

[Star18] Starship Technologies: *Starship Robot*. URL https://www.starship.xyz/. - abgerufen am 2018-08-01

[Stil18] Still GmbH: *Still Automatisierungslösungen*. URL https://www.still.de/intralogistik-systeme/automatisierungsloesungen.html. - abgerufen am 2018-08-01

[SüRo14] Süße, Herbert; Rodner, Erik: *Bildverarbeitung und Objekterkennung: Computer Vision in Industrie und Medizin*. Wiesbaden : Springer Vieweg, 2014 - ISBN 978-3-8348-2605-3

[Ullr18] Ullrich, Günther: Erwartungen an die FTS-Branche – Technologie und Innovation. In: *VDI-Fachkonferenz Agile Produktionsversorgungssysteme*, *VDI-Berichte*. Bd. 2325. München, 2018

[Velo18] Velodyne LiDAR: *Products*. URL http://www.velodynelidar.com/products.html. - abgerufen am 2018-08-01

[ZhTu17] Zhou, Yin; Tuzel, Oncel: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In: *arXiv:1711.06396 [cs]* (2017)

[ZSKF15] Zhirong Wu; Song, Shuran; Khosla, Aditya; Fisher Yu; Linguang Zhang; Xiaoou Tang; Xiao, Jianxiong: 3D ShapeNets: A deep representation for volumetric shapes. In: *Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)* : IEEE, 2015 - ISBN 978-1-4673-6964-0, S. 1912–1920

Article is protected by German copyright law

DOI: 10.2195/lj_Proc_thiel_de_201811_01
URN: urn:nbn:de:0009-14-47541

Marko Thiel, M.Sc., Wissenschaftlicher Mitarbeiter am Institut für Technische Logistik, Technische Universität Hamburg. Marko Thiel studierte bis 2015 Maschinenbau und Theoretischen Maschinenbau an der Technischen Universität Hamburg.

Dr. Johannes Hinckeldeyn, Oberingenieur am Institut für Technische Logistik, Technischen Universität Hamburg. Nach seiner Promotion in Großbritannien war Johannes Hinckeldeyn als Chief Operating Officer für einen Hersteller von Mess- und Labortechnik für die Batterieforschung tätig. Johannes Hinckeldeyn studierte Wirtschafsingenieurwesen, Produktionstechnik und -management in Hamburg und Münster.

Prof. Dr.-Ing. Jochen Kreutzfeldt, Professor und Leiter des Instituts für Technische Logistik, Technischen Universität Hamburg. Nach seinem Maschinenbaustudium mit der Vertiefung Produktionstechnik war Jochen Kreutzfeldt in verschiedenen leitenden Positionen bei einem Unternehmen für Automobilsicherheitstechnik tätig. Anschließend übernahm Jochen Kreutzfeldt eine Professur für Logistik an der Hochschule für Angewandte Wissenschaften Hamburg und wurde Leiter des Instituts für Produkt- und Produktionsmanagement.

Adresse: Institut für Technische Logistik, Technische Universität Hamburg, Theodor-Yorck-Straße 8, 21079 Hamburg, Deutschland; Telefon: +49 40 42878-3422, E-Mail: marko.thiel@tuhh.de

Article is protected by German copyright law