

Praktikum4_v03tt

January 17, 2024

1 Praktikum Intelligente Sensortechnik 4

Tim Tiedemann, Thomas Lehmann, Tobias De Gasparis

Version 09.01.2024

2 Einfache intelligente Sensoren und Datenvorverarbeitung

Im Praktikum 4 geht es um die Verarbeitung hochdimensionaler Daten bzw. das Clustern der Daten.

Lesen Sie sich die Aufgaben gut durch. Sollten Sie eine Aufgabe nicht lösen können, so beschreiben Sie zumindest, wie weit Sie gekommen sind und auf welche Weise Sie vorgegangen sind.

Beachten Sie auf der methodischen Seite, dass die Schritte der Datenerhebung, der Datenauswertung und der Kommentierung des Ergebnisses ausgeführt werden. Alle Diagramme sind korrekt zu beschriften.

Die Aufgaben sind direkt hier als Protokoll zu bearbeiten. Das abgegebene Notebook soll ausführbar sein. Daneben ist der PDF-Export des Notebook mit abzugeben.

Autoren des Protokolls: Haron Nazari, Anton Tchekov

3 Hintergrund

Intelligente Sensoren sollen nicht nur Daten erfassen, sondern direkt eine Klassifikation mit Informationen auf abstrakterer Ebene aus der Umwelt liefern. Beispielsweise soll eine Geste oder Gegenstände erkannt werden und diese Information über die Schnittstelle zu weiteren Systemkomponenten bereitgestellt werden. In diesem Praktikum wird die Leistungsfähigkeit der Klassifikation einfacher Algorithmen des nicht-überwachten Machine Learnings für diese Aufgabe untersucht.

4 Vorbereitungsaufgaben

4.1 Sensoren mit hochdimensionaler Ausgabe von Daten

Besorgen Sie sich Informationen zu den folgenden Sensoren:

- Hokuyo URG-04LX-UG01
- Hokuyo UTM-30LX-EW
- Velodyne VLP-16

Um was für einen Sensor handelt es sich jeweils und welches Messprinzip wird verwendet? Was wird gemessen? Was sind hier der Messbereich, die Auflösung und die Sample-/Messrate?

Über welche Schnittstelle(n) können die Messdaten ausgegeben werden? Wie werden jeweils Betriebsspannung und Datenleitungen angeschlossen?

URG-04LX

Der URG-04LX ist ein Laserscanner, welcher einen Bereich scannt. Der Laser nutzt einen Infrarotsensor und das Sichtfeld beim Scan ist 240 Grad. Er kann Distanzen bis zu 4000 mm messen und die maximale Fehlerrate beträgt 40 mm bei 4000 mm Entfernung. Das Messprinzip basiert auf der Phasendifferenz.

Distanz: Bis zu 4000 mm

Auflösung: 1 mm

Genauigkeit:

± 1 % des Messwertes bei 1000 - 4000 mm






± 10 mm bei 60 - 1000mm

Geschwindigkeit: 100 ms / Scan

Schnittstellen: RS-232, USB

Betriebsspannung: 5 VDC

Pinout von Sensor:

	URG-04LX	Lead Color
1	N.C.	RED 
2	N.C.	WHITE 
3	OUTPUT (SYNCHRONOUS)	BLACK
4	GND (9pin Dsub 5p)	PURPLE 
5	RxD (9pin Dsub 3p)	YELLOW 
6	TxD (9pin Dsub 2p)	GREEN 
7	0V	BLUE
8	DC 5V	BROWN

UTM-30LX-EW

Der UTM-30LX-EW nutzt eine Laserquelle um für jeden Schritt die Distanz zu messen. Ein Schritt ist 0.25 Grad groß.

Distanz: 0.1 - 30 m (60 m Maximum nicht garantiert)

Auflösung: 1 mm

Genauigkeit:

0.1 - 10m: ± 30 mm

10 - 30 m: ± 50 mm

Sichtfeld: 270 Grad**Schrittweite:** 0.25 Grad**Schnittstellen:** Ethernet / Synchronous Output**Geschwindigkeit:** 25 ms**Velodyne VLP-16**

Der VLP-16 LiDAR Sensor kann über Ethernet (TCP/IPv4) angesteuert werden. Der Sensor verwendet Time-of-Flight als Messprinzip.

Die Betriebsspannung ist von 8 bis 19 Volt. Stromverbrauch 0.9 A (8 Watt) in normaler Operation (maximal 3 A).

Die Konfiguration des Sensors kann über ein Web-Interface erfolgen oder mit HTTP-Requests an API-Endpunkten.

5 Im Labor

5.1 Objektidentifikation in Daten von Laserscannern

Mit LiDARs vom Typ Hokuyo URG-04LX-UG01 sollen Sie die Umgebung der Sensoren erfassen und mittels Clustering-Algorithmus auswerten. Sie sollen untersuchen, ob die Objekte der Umgebung durch den Cluster-Algorithmus gefunden werden.

5.1.1 Datenerfassung

Verwenden Sie in Absprache mit dem Laborbetreuer einen Sensor vom Typ Hokuyo URG-04LX-UG01 für die Datenerfassung. Wie werden Daten erfasst und wie können sie auf diese zugreifen?

TODO: Im Labor Daten vom Sensor erfassen.

Sammeln Sie Daten von einem Scan und versuche Sie die Daten in einer zweidimensionalen Karte darzustellen. Welche Form von Koordinatensystem bietet sich hier an und wie können Sie so einen Plot erzeugen? Können Sie Objekte in der Umgebung identifizieren?

Ein polares Koordinatensystem bietet sich an, da der Sensor Entfernungen in einem bestimmten Winkel liefert, die von einem zentralen Punkt aus gemessen wurden.

2D-Karte der Sensordaten von einem Scan:

TODO: Bild generieren mit echten Sensordaten

TODO

TODO: Welche Objekte sind erkennbar?

Python code:

[]: `TODO: Paste Python code here`

Nehmen Sie von zwei LiDARs einen Scan auf. Gehen Sie davon aus, dass einer der Laserscanner im Ursprung eines kartesischen Koordinatensystems (Weltkoordinatensystem) angebracht ist und mit der Mittelachse genau in Richtung der y-Achse ausgerichtet ist. Fügen Sie die Daten des zweiten Laserscanners mittels geeigneter Rotation und Translation (Argumente experimentell bestimmen) in das Weltkoordinatensystem ein, so dass sich ein Gesamtbild in der Karte ergibt. Optional: Geben Sie die nötige Rotationsmatrix und den Translationsvektor an bzw. bauen Sie Ihre Datentransformation so auf, dass mit einer Rotationsmatrix und einem Translationsvektor gearbeitet wird.

Da wir aber zwei Sensoren verwenden, ist es sinnvoll, die Werte aus dem polaren Koordinatensystem in ein kartesisches Koordinatensystem zu überführen, damit die Berechnungen für die Translation und Rotation des zweiten Sensors einfacher sind.

TODO: Werte ausfüllen

Experimentell bestimmter Translationsvektor:

[0, 0, 0]

Rotationswinkel:

0 °

Rotationsmatrix:

[
[0, 0],
[0, 0],
]

TODO: Scan Gesamtbild beider Sensoren

Gesamtbild 2D-Karte mit Werten beider Sensoren:

TODO

Python code:

[]: TODO: Paste Python code here

5.1.2 Clustering der Daten

Führen Sie nun ein Clustering mittels DBSCAN aus der scikit-learn-Bibliothek auf den gesammelten LiDAR-Daten von einem Sensor durch. Verwenden Sie evtl. erstmal nur einen Teil, z.B. die ersten 100 Sensordimensionen (Entfernungswerte). Finden Sie geeignete Parameter “min_samples” und “eps”. Stellen Sie die Cluster-Zuordnung der Punkte in der Karte dar.

TODO

Wieviele Cluster werden in Ihrem Datensatz identifiziert? Welche Objekte könnten es sein?

TODO: Cluster beschreiben, Objekte identifizieren

Sollte im LiDAR-Datensatz eine Standardisierung durchgeführt werden? Was spricht dafür und was dagegen? Beachten Sie die Arbeitsweise von DBSCAN.

Es ist eher nicht sinnvoll, eine Standardisierung durchzuführen, da alle Entfernungsmesswerte die gleiche Einheit haben, und die absolute Entfernungsinformation (z.B. in Metern) wichtig ist und bei der Standardisierung verloren gehen würde. Zudem ist DBSCAN resistent gegenüber Ausreißern, da es dichtebasiert arbeitet, weswegen eine Standardisierung nicht notwendig wäre.

Können Sie eine Daumenregel für die Wahl der Parameter von DBSCAN ableiten, so dass Objekte in der Umgebung als getrennte Objekte erkannt werden?

DBSCAN benötigt zwei Eingabeparameter, Minimum samples (MinPts) und Epsilon (eps).

Für MinPts kann man als Faustregel $2 * \text{dim}$ wählen, wobei dim die Anzahl der Dimensionen des Datensatzes ist. Dabei sollte MinPts mindestens 3 sein. Je größer der Datensatz ist, desto größer sollte man MinPts wählen.

eps kann man mithilfe eines k-distance-graph bestimmen, indem man $k = \text{MinPts}$ setzt und eps auf den Wert setzt, bei dem der Graph einen starken Knick zeigt.

Python code

[]: TODO: Paste python code here

5.2 Bewegungserkennung

Aus den Daten der Sensoren des Nucleo-Moduls soll nun eine Bewegung identifiziert werden.

5.2.1 Datenerfassung

Verwenden Sie im folgenden das IKS01A3-Board und Ihr Programm, wie Sie es zum Sammeln der 12-dimensionalen 1024-Sample Datensätzen verwendet haben.

Nehmen Sie einen Datensatz auf, in dem das Modul zunächst still in einer Position (und Ausrichtung) steht, dann eine Bewegung durchgeführt wird (z.B. Translation und dann Rotation) und das Modul dann still in einer anderen Position (und Ausrichtung) steht (alles innerhalb eines 1024-Sample-Datensatzes). Dokumentieren (Skizze) Sie die Bewegung! (Empfehlung: Filmen Sie die Bewegung, damit Sie ggf. einige Punkte in den Datensätzen bei der Auswertung besser einer Bewegung zuordnen können.)

Optional können Sie versuchen, zusätzlich einen Datensatz mit drei Ruhephasen innerhalb der 1024 Samples aufzunehmen.

5.2.2 Auswertung

Stellen Sie sich diesen Datensatz graphisch als Plot (2D/3D) dar und prüfen Sie, ob die verschiedenen Phasen zu erkennen sind. Zeigen Sie hier einen geeigneten Plot/Ausschnitt.

TODO: Bild erzeugen und plotten

Plots des Datensatzes:

TODO

Versuchen Sie mittels Clustering-Algorithmus (k-Means und DBSCAN) die Bewegungen/Bewegungsabschnitte in den Daten zu identifizieren. Welches Verfahren kann wie gut die Bewegung/Bewegungsabschnitte bei welchen Parametern identifizieren? Welche Normierung/Skalierung war sinnvoll? Ist eine Reduktion mit PCA sinnvoll? Stellen Sie die Ergebnisse des Clusterings ggf. auch in Plots über die Zeit dar.

Fügen Sie hier Ihre Analyse kommentiert ein.

TODO

5.2.3 Sensorsystem

Wenn Sie nun das Sensorboard zusammen mit den Cluster-Algorithmen als Sensorsystem betrachten, welche Information(en) könnte die Schnittstelle des Sensorsystems

als Ergebnis des Clusters bereitstellen? Was wäre bei verschiedenen Bewegungen (2 Ruhephasen, 3 Ruhephasen) eine mögliche Ausgabe?

TODO:

Die Schnittstelle könnte einen Zeitabschnitt zurückliefern, der mit `t_start` und `t_end` definiert ist, zusammen mit einer Beschreibung, was in diesem Zeitabschnitt für eine Bewegung stattgefunden hat.

z.B.:

t_start	t_end	Art der Bewegung
0.00	1.00	Ruhephase
1.00	1.50	Bewegung entlang +X und -Z Achse
1.50	2.00	Drehung um Y Achse
2.00	3.00	Ruhephase